Comparison of a Best and a Near—Best Basis Approach to Wavelet Analysis of Electrocardiograms

by

Robert Clifford Bruce Steacy

B.Sc., University of British Columbia, 1972.

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in the Department of Mathematics and Statistics.

We accept this thesis as conforming to the required standard.

[redacted]

Dr. R. Illner, Supervisor (Department of Mathematics & Statistics)

[redacted]

Dr. D. Hewgill, Departmental Member (Department of Mathematics & Statistics)

[redacted]

Dr. N. Horspool, Outside Member (Department of Computer Science)

[redacted]

Dr. V. Bhargava, External Examiner (Department of Electrical and Computer Engineering)

Supervisor: Dr. R. Illner.

Abstract

A necessary first step in the high—speed computerized analysis of electrocardiograms is to transform the data so that as much of the energy of the signal is contained in as few coefficients as possible. This facilitates both de—noising and data compression, so that further steps such as arrhythmia detection can then be carried out.

A logical starting point for electrocardiogram research is provided by the wavelet techniques contained in the standard reference work *Numerical Recipes in C*, by Press et al. [10]. These are based on the work of Ingrid Daubechies (compactly supported wavelets) and S. Mallat (the pyramid algorithm).
Using a well—known measure of the expected information of a signal, this thesis compares the basic treatment of electrocardiograms to two improved schemes, the Near—Best—Basis of Taswell, and the Best Basis of Coifman and Wickerhauser. Mathematical definitions and foundations for "Best" and "Near—Best" are given in Chapter 3 of the thesis.

There are two conclusions. The Near—Best—Basis of Taswell gives a significant improvement over the basic pyramid algorithm treatment for low additional computational cost. The Best Basis produces a further improvement over the Near—Best—Basis which is negligible for practical purposes, and at a large increase in computational cost. These conclusions have valuable practical implications for researchers in this field, and for the estimated several hundred million people worldwide who face serious heart problems in the next decade.

Examiners:

[redacted]

Dr. R. Illner, Supervisor (Department of Mathematics & Statistics)

[redacted]

Dr. D. Hewgill, Departmental Member (Department of Mathematics & Statistics)

[redacted]

Dr. N. Horspool, Outside Member (Department of Computer Science)

[redacted]

Dr. V. Bhargava, External Examiner, (Department of Electrical and Computer Engineering)

Contents

List of Tables

List of Figures

Acknowledgements

I wish to express my appreciation to my supervisor Dr. Reinhard Illner for his guidance during the completion of this thesis. I also wish to thank department member Dr. Denny Hewgill and outside member Dr. Nigel Horspool for their constructive comments. I wish to express my appreciation to Dr. Vijay Bhargava of Electrical and Computer Engineering, University of Victoria, for readily agreeing to serve as my external examiner.

Special thanks go to the present Chair of the Department of Mathematics and Statistics, Dr. Bill Pfaffenberger, and the former Chair, Dr. David Leeming, for providing an atmosphere in which meaningful and useful research can be accomplished.

Of course, this thesis could not have been completed without the understanding and support of my wife Nancy, and children Erin and Sean.

I am grateful for the encouragement of my father, Charles Steacy, and my late mother, Margaret Steacy, to whom this thesis is dedicated.

Dedication

To my mother

Chapter 1

Introduction

The electrocardiogram (ECG) is routinely performed on almost every hospitalized adult. Its usefulness in modern medicine cannot be overstated. Heart disease and stroke remain the number one killer of both men and women in Canada, according to the Heart and Stroke Foundation of Canada.

Researchers wishing to subject electrocardiograms to high-speed computerized analysis have a number of requirements and considerations which must be met. To begin with, the data may be sufficiently noisy that it is difficult to identify the salient features of the heartbeats, and denoising of the signal is the first priority. It may be that data storage is limited, and it becomes crucial to be able to compress data. Lastly, as a normal 24—hour recording contains nearly one hundred thousand beats, there is a need for a rapid technique to identify abnormal beats, which are known as arrhythmias.

Figure 1.1: Sample 1 shows a noisy recording of a typical heartbeat. In the middle of the recording, there is a prominent upward spike, which together with the smaller downward spikes immediately adjacent to it, form the q, r, s—*complex*. Just before and just after the q, r, s—*complex*, respectively, we find

two local maxima known as the *p—wave* and the *t—wave*.

The new branch of mathematics known as wavelet analysis has found increasing acceptance for ECG applications. Two of the main features of wavelet analysis which recommend it for this purpose are localization, and concentration of data from a large number of points primarily into a small number of coefficients.

Localization means that our analysis of one part of the ECG is not unduly influenced by other parts of it. The concentration of data means that we can invoke techniques involving the setting of most of the coefficients in the wavelet transform of the signal to zero. Precisely which coefficients are set to zero is a topic known as threshholding, which has a large and growing literature associated with it. These techniques find application in both the de—noising and the data compression referred to above. Also, the wavelet transform executes significantly faster on a computer than the Fast Fourier transform.

A starting point for the wavelet analysis of ECG's is provided by the standard reference *Numerical Recipes in C*, by Press et al. [10]. Here, one finds the Discrete Wavelet Transform (DWT), also referred to as a *pyramidal algorithm.*

Recognizing the desirability of compressing as much of the power of the signal into as few coefficients of the wavelet transform as possible, we shall use a well-known objective measure of this feature known as the *Shannon entropy* of the signal. Shannon entropy is a measure of the expected information content of a signal, and lower Shannon entropy therefore translates into better compression.

This thesis compares the Shannon entropy of the wavelet transform of a single heartbeat using the DWT with that of two improved methods. These are the Near—Best—Basis of Taswell [15], and the Best Basis of Coifman and Wickerhauser [2], [3].

It is found that the Near—Best—Basis of Taswell provides a significant reduction, i.e., improvement, in Shannon entropy over the basic DWT, and that this is accomplished with low additional computational cost. The Best Basis of Coifman and Wickerhauser, on the other hand, is found to provide an improvement over the Near—Best—Basis which is negligible for practical purposes, and does so at a great increase in computational cost.

The conclusion of this thesis is that, of the three techniques, the Near—Best—Basis of Taswell is to be preferred over the other two.

1.1 Background

Born in Auxerre, France, in 1768, Jean Joseph Baptiste Joseph Fourier started his career as an administrator. When Napoleon invaded Egypt in 1798, Fourier advised on engineering and diplomatic matters. He became a prominent Egyptologist.

Fourier returned to France in 1801 and served as an administrator at Grenoble until 1814, during which time he contributed to journals of Egyptology. In 1807, he began his investigation into the conduct of heat, completing his work in Paris and publishing his results in 1822.

Fourier showed that the functions which described the flow of heat in solid bodies could be broken down into components whose equations were sines and

cosines.[11]

The Fourier series of a 2π—periodic function is

$$f(x) = a_0 + \sum_{k=1}^{\infty} (a_k \cos kx + b_k \sin kx)$$

with

$$a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\ dx$$

$$a_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\ \cos kx\ dx$$

$$b_k = \frac{1}{\pi} \int_{-\pi}^{\pi} f(x)\ \sin kx\ dx$$

Today it is more customary to write

$$f(x) = \sum_{k=-\infty}^{\infty} c_k\, e^{ikx}$$

with

$$c_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x)\ e^{-ikx}\ dx$$

We shall require Parseval's Formula

$$\int_{-\pi}^{\pi} |f(x)|^2\ dx = 2\pi \sum_{k=-\infty}^{\infty} |c_k|^2.$$

Conditions for existence and convergence of Fourier series are well known and we shall not present them here.

Many problems in both pure and applied mathematics are solved by the following technique: transform the problem from its present domain, where the steps are difficult, to another domain, where they are simpler. Solve the

problem, and then transform the solution back to the original domain. Logarithms, Fourier transforms, Laplace transforms, and wavelet transforms are all examples of this approach to problem solving.

For well over a century, the usefulness of the Fourier transform was hampered by its computational complexity. The Discrete Fourier transform is a relatively expensive operation; i.e., $O(n^2)$, where n is the number of samples in each signal.

In a 1965 paper in *Mathematics of Computation*, J.W. Cooley and J.W. Tukey realized that the Discrete Fourier transform could be accelerated. The result was the Fast Fourier transform, or FFT, whose cost is reduced to $O(n \log n)$. For vectors of even modest length, this is a substantial reduction in time and cost of computing. The result was an explosion of applications of Fourier analysis.

Fourier analysis has one major drawback, however, and that is that sines and cosines do not have compact support; i.e., the functions are not zero outside of some compact set. This results in poor localization. Various windowing schemes were devised to overcome this difficulty, resulting in a substantial literature.

A detailed discussion of wavelets will be presented in the next section. For now, it is sufficient to mention that the compactly supported wavelets constructed by Ingrid Daubechies in 1988 overcame the problems caused by lack of compact support which limited the usefulness of Fourier analysis.

Compact support permits excellent localization, that is, the wavelet transform can represent and then analyze local behavior of a signal. Also, the wavelet transform cost is reduced still further to $O(n)$.

1.2 Necessary Terminology

This section sets the terminology and symbols which will be used throughout the thesis.

For the reader who is already quite familiar with wavelets, a quick scan of this section will confirm the precise usage which we make of particular terms and symbols.

Readers, such as engineers, who have a good background in Fourier analysis, might regard this section as a useful introduction to the topic of wavelets.

Definition 1.1 *The Fourier transform* $\hat{f}(\omega) = (\mathcal{F}f)(\omega)$ *of a function* $f(t)$ *is*

$$\hat{f}(\omega) = (\mathcal{F}f)(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i\omega t} f(t) dt \tag{1.1}$$

where ω *is the variable in the frequency domain. This naturally leads to the Inverse Fourier transform* $f(t) = (\mathcal{F}^{-1}\hat{f})(\omega)$ *of the Fourier transform* $\hat{f}(\omega)$*, which is*

$$f(t) = (\mathcal{F}^{-1}\hat{f})(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{i\omega t} \hat{f}(\omega) d\omega \tag{1.2}$$

Theorem 1.1 *Convolution Theorem*

$$Let \ (f * g)(t) = \int_{-\infty}^{\infty} f(t-\tau) g(\tau) d\tau \tag{1.3}$$

$$Then \ \widehat{(f * g)}(\omega) = \sqrt{2\pi} \hat{f}(\omega) \cdot \hat{g}(\omega) \tag{1.4}$$

We shall require the Differential Operators to Algebraic Operators transform

$$\mathcal{F}(\frac{d^\ell}{dx^\ell}f) = (i\omega)^\ell(\mathcal{F}f)(\omega) \tag{1.5}$$

Definition 1.2 *The Schwartz space* S

$$\mathsf{S}(\mathbb{R}) = \{f \in C^\infty(\mathbb{R})|\ \forall\ m,n \in \mathbb{N}\ \sup_{x\in\mathbb{R}}|(1+|x|)^n f^{(m)}(x)| < \infty\} \tag{1.6}$$

Note that if we restrict functions to the Schwartz space S, then we are guaranteed that not only will the Fourier transform $\hat{f}(\omega)$ exist, but that the Inverse Fourier transform will give us back $f(x)$.

Lemma 1.1 *Parseval's Identity*

Let $\phi,\ \psi \in \mathsf{S}$ *(the Schwartz space).*

$$\textit{Then}\int_{-\infty}^{\infty}\phi(t)\ \overline{\psi(t)}\ dt = \int_{-\infty}^{\infty}\hat{\phi}(\omega)\ \overline{\hat{\psi}(\omega)}\ d\omega \tag{1.7}$$

Lemma 1.2 *Plancherel's Formula*

$$\int_{-\infty}^{\infty}|f(x)|^2dx = \int_{-\infty}^{\infty}|\hat{f}(\omega)|^2d\omega \tag{1.8}$$

Definition 1.3 *A complex Hilbert Space* $\mathbb{H}$ *is a vector space over* $\mathbb{C}$ *which possesses a scalar product* $<,>: \mathbb{H}\times\mathbb{H} \longrightarrow \mathbb{C}$. *As in [5] we will use a scalar product which is linear in the first argument.*

$$< \lambda_1 u_1 + \lambda_2 u_2, v > = \lambda_1 < u_1, v > + \lambda_2 < u_2, v > \tag{1.9}$$

$$< u, \lambda_1 v_1 + \lambda_2 v_2 > = \overline{\lambda_1} < u, v_1 > + \overline{\lambda_2} < u, v_2 > \tag{1.10}$$

$$< v, u > = \overline{< u, v >} \tag{1.11}$$

$$< u, u > \geq 0 \ \forall u \in \mathbb{H}, with < u, u > > 0 \ if \ u \neq 0 \tag{1.12}$$

$$The \ norm \ \|u\| \ of \ u \ is \ \|u\|^2 =< u, u > \tag{1.13}$$

In a Hilbert Space, $\|u\| = 0$ implies $u = 0$. All sequences in $\mathbb{H}$ which are Cauchy sequences with respect to $\|\cdot\|$ have their limit in $\mathbb{H}$, i.e. a Hilbert Space is a complete scalar product space.

The Hilbert Spaces which are of primary interest to us are $\mathbb{L}^2(\mathbb{R})$, $\mathbb{L}^2[0, 2\pi]$ and $\ell^2(\mathbb{Z})$, with the following inner products:

$\mathbb{L}^2(\mathbb{R})$

$$< f, g >= \int_{-\infty}^{\infty} f(x) \ \overline{g(x)} \ dx \tag{1.14}$$

$\mathbb{L}^2[0, 2\pi]$

$$< f, g >= \int_{0}^{2\pi} f(x) \ \overline{g(x)} \ dx \tag{1.15}$$

$\ell^2(\mathbb{Z})$

$$< c, d >= \sum_{n=-\infty}^{\infty} c_n \overline{d_n} \tag{1.16}$$

Although these are infinite—dimensional Hilbert Spaces, they possess countable orthonormal bases. Such Hilbert Spaces are called separable.

A standard inequality in a Hilbert Space is the Cauchy—Schwartz inequality:

$$| < v, w > | \leq ||v|| \cdot ||w|| \tag{1.17}$$

We shall also make use of an item of terminology borrowed from signal processing, and refer to the $\mathbb{L}^2(\mathbb{R})$ inner product of a function with itself, or the $\ell^2(\mathbb{Z})$ inner product of an element of ℓ^2 with itself, as the *energy* of the object in question.

We now proceed to a discussion of the Haar and Daubechies D4 wavelets.

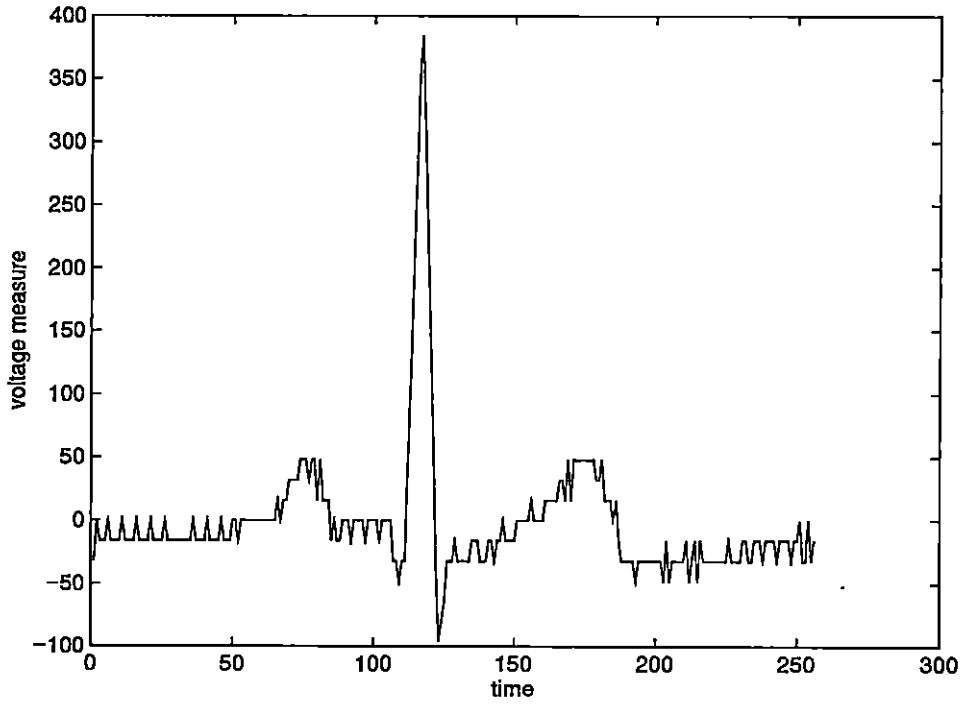


Figure 1.1: Sample 1

Time is in units of $\frac{1}{250}$ second. Voltage measure is particular to the ambulatory recording device being used and does not correspond exactly to a 12—lead ECG machine such as might be used in a hospital.

Chapter 2

The Haar and Daubechies D4 Wavelets

2.1 The Haar Wavelet (Haar, 1910)

Let

$$h_0(x) = \psi_0(x) = \chi_{[0,1)}$$

the characteristic function of the half—open interval $[0, 1)$.

We can also express this as

$$\psi_0(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & otherwise \end{cases}$$

Now let

$$h_1(x) = \psi_1(x) = \psi_0(2x) - \psi_0(2x - 1)$$

This can be expressed as

$$\psi_1(x) = \begin{cases} 1 & 0 \leq x < \frac{1}{2} \\ -1 & \frac{1}{2} \leq x < 1 \\ 0 & otherwise \end{cases}$$

Now we consider the family of functions

$$\psi_{m,n}(x) = 2^{-\frac{m}{2}}\psi_1(2^{-m}x - n) \qquad m, n \in \mathbb{Z}$$

The proof that the $\psi_{m,n}(x)$ constitute an orthonormal basis for $\mathbb{L}^2(\mathbb{R})$ may be found in [5], pp. 10—13.

In order to represent a function $f(x)$ in the $\psi_{m,n}(x)$ basis, we form $\mathbb{L}^2(\mathbb{R})$ inner products of $f(x)$ with the $\psi_{m,n}(x)$. We then have

$$f(x) = \sum_{m,n} < f(x), \psi_{m,n}(x) > \ \psi_{m,n}(x)$$

If we introduce the notation

$$\mathbb{T}^{wav}_{m,n}(f) = a_o^{-\frac{m}{2}} \int_{-\infty}^{\infty} f(t)\ \psi_1(a_0^{-m}t - nb_0)dt$$

we have, in the case of the Haar basis, $a_0 = 2$ and $b_0 = 1$ and so

$$\mathbb{T}^{wav}_{m,n}(f) = 2^{-\frac{m}{2}} \int_{-\infty}^{\infty} f(t)\ \psi_1(2^{-m}t - n)dt$$

and so

$$f(x) = \sum_{m,n} \mathbb{T}^{wav}_{m,n}(f)\ 2^{-\frac{m}{2}}\ \psi_1(2^{-m}x - n)$$

or more concisely

$$f(x) = \sum_{m,n} \mathbb{T}^{wav}_{m,n}(f)\ \psi_{m,n}(x)$$

2.2 Multiresolution Analysis

Daubechies [5] points out that the approximation of a function in $\mathbb{L}^2(\mathbb{R})$ with the Haar basis implicitly uses a "multiresolution" approach. The difference between the approximation with resolution 2^{j-1} and the next coarser level, with resolution 2^j, is a linear combination of the $\psi_{m,n}(x)$.

The Haar wavelet representation of $f(x)$ depends on a coarse level with support width 2^j, where $j \in \mathbb{Z}$, and finer levels with support width 2^{j-1}, 2^{j-2}, 2^{j-3}, ...

We make use of a ladder of subspaces (V_j), $j \in \mathbb{Z}$, where

$V_j = \{f \in \mathbb{L}^2(\mathbb{R});$ f piecewise constant on $[\ 2^j k,\ 2^j(k+1)\ ),\ k \in \mathbb{Z}\ \}$.
These spaces have the following properties:

$$\ldots \subset V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset \ldots \tag{2.1}$$

$$\bigcap_{j\in\mathbb{Z}} V_j = \{0\},\ \overline{\bigcup_{j\in\mathbb{Z}} V_j} = \mathbb{L}^2(\mathbb{R}); \tag{2.2}$$

$$f \in V_j \leftrightarrow f(2^j\cdot) \in V_0; \tag{2.3}$$

$$f \in V_0 \rightarrow f(\cdot - n) \in V_0 \quad \forall n \in \mathbb{Z} \tag{2.4}$$

$$\exists\ \phi \in V_0\ \ni\ \phi_{0,n} = \phi(x-n)$$

$$\textit{constitute an orthonormal basis for } V_0. \tag{2.5}$$

Daubechies [5] gives the following explicit recipe for the construction of ψ, the "mother" wavelet:

Since the scaling function or the "father" wavelet $\phi \in V_0 \subset V_{-1}$ and the $\phi_{-1,n}(x) = \sqrt{2}\phi(2x-n)$ constitute an orthonormal basis for V_{-1} by

the properties of the V_j above, then there exist $\alpha_n = \sqrt{2} < \phi, \phi_{-1,n} >$ so that $\phi(x) = \sum_n \alpha_n \phi(2x - n)$ (the Dilation Equation). We then take $\psi(x) = \sum_n (-1)^n \alpha_{-n+1} \phi(2x - n)$.

In the case of the Haar wavelets, we have $\phi = \chi_{[0,1)}$. $\psi_{m,n}(x) = 2^{-\frac{m}{2}} \psi(2^{-m}x - n)$, so $\phi_{m,n}(x) = 2^{-\frac{m}{2}} \phi(2^{-m}x - n)$. This gives $\phi_{0,n}(x) = \phi(x - n)$ and $\phi_{-1,n}(x) = \sqrt{2}\phi(2x - n)$.

Accordingly,

$$< \phi, \phi_{-1,n} >= \frac{\sqrt{2}}{2} \; if \; n = 0 \; or \; n = 1 \; or \; 0 \; otherwise. \tag{2.6}$$

This gives $\alpha_n = \sqrt{2} \cdot \frac{\sqrt{2}}{2} = 1$ if n = 0 or 1, $\alpha_n = 0$ for all other values of n.

It is clear that the Haar father wavelet satisfies the Dilation Equation $\phi(x) = \sum_n \alpha_n \phi(2x - n) = \phi(2x) + \phi(2x - 1)$.

We then have for the Haar mother wavelet $\psi(x) = \sum_n (-1)^n \alpha_{-n+1} \phi(2x - n) = \alpha_1 \phi(2x) - \alpha_0 \phi(2x - 1) = \phi(2x) - \phi(2x - 1)$.

2.3 Conditions on the α_n

It is customary to normalize $\phi(x)$ so that $\int_{-\infty}^{\infty} \phi^2(x) \; dx = 1$.

Definition 2.1 *Condition 1*

From

$$\phi(x) = \sum_{n \in \mathbb{Z}} \alpha_n \; \phi(2x - n) \tag{2.7}$$

it follows that

$$\int_{-\infty}^{\infty} \phi(x)\ dx = \int_{-\infty}^{\infty} \sum_{n\in\mathbb{Z}} \alpha_n\ \phi(2x-n)\ dx \tag{2.8}$$

$$= \frac{1}{2}\sum_{n\in\mathbb{Z}} \alpha_n \int_{-\infty}^{\infty} \phi(2x-n)\ d(2x-n) \tag{2.9}$$

But since $\int_{-\infty}^{\infty} \phi(x)\ dx = 1$ *we have that* $\int_{-\infty}^{\infty} \phi(2x-n)\ d(2x-n) = 1\ \forall n$ *and therefore*

$$1 = \frac{1}{2}\sum_{n\in\mathbb{Z}} \alpha_n \tag{2.10}$$

$$\textit{i.e. that } \sum_{n\in\mathbb{Z}} \alpha_n = 2\ (\textit{Condition } 1) \tag{2.11}$$

Definition 2.2 *Condition 2: (also known as Condition O — Orthogonality)*

$$\sum_{n\in\mathbb{Z}} \alpha_n\ \alpha_{n-2m} = 2\delta_{0m}\ \ \forall m \in \mathbb{Z} \tag{2.12}$$

where δ_{mn} *is the Kronecker delta.*

Why is this condition desirable? It guarantees the following useful Lemma:

Lemma 2.1 *Suppose that* ϕ_0 *is a function such that* $\{\phi_0(2x-n)\}_{n\in\mathbb{Z}}$ *is an orthogonal set in* $\mathbb{L}^2(\mathbb{R})$. *If Condition O holds, then the translates of* $\phi_1(x) = \sum_{n\in\mathbb{Z}} \alpha_n\ \phi_0(2x-n)$ *are also orthogonal. Further, by our convention that* $\int_{-\infty}^{\infty} \phi^2(x)\ dx = 1$, *we have that the translates are also orthonormal.*

Proof. We compute

$$\int_{-\infty}^{\infty} \phi_1(x)\ \phi_1(x-m)\ dx \tag{2.13}$$

$$= \int_{-\infty}^{\infty} \sum_{k\in\mathbf{Z}} \alpha_k \, \phi_0(2x-k) \cdot \sum_{\ell\in\mathbf{Z}} \alpha_\ell \phi_0(2x-2m-\ell) \, dx \tag{2.14}$$

$$= \sum_{k\in\mathbf{Z}} \alpha_k \, \alpha_{k-2m} \int_{-\infty}^{\infty} \phi_0^2(2x-k) \, dx \tag{2.15}$$

where we have used that if $\ell \neq k-2m$, $\int_{-\infty}^{\infty} \phi_0(2x-k) \, \phi_0(2x-2m-\ell) \, dx = 0$. Now note that by substituting $z = 2x-k$, $dz = 2\,dx$, (2.16) is equal to

$$\frac{1}{2} \sum_{k\in\mathbf{Z}} \alpha_k \, \alpha_{k-2m} \int_{-\infty}^{\infty} \phi_0^2(z) \, dz \tag{2.16}$$

$$= 0 \; if \; m \neq 0 \; since \sum_{k\in\mathbf{Z}} \alpha_k \, \alpha_{k-2m} = 0 \; if \; m \neq 0. \tag{2.17}$$

Definition 2.3 *Condition 3 (also known as Condition A — Approximation)*

For $p \in \mathbf{N}$

$$\sum_{n\in\mathbf{Z}} (-1)^n \, n^m \, \alpha_n = 0 \;\; for \; m = 0, 1, \ldots, p-1 \tag{2.18}$$

This condition guarantees that the polynomials $1, \; x, \; x^2, \; \ldots, \; x^{p-1}$ *are linear combinations of the translates* $\phi(x-n)$. *The first* p *moments of the wavelet* $\psi(x)$ *associated with* ϕ *are then zero:*

$$\int_{-\infty}^{\infty} x^m \, \psi(x) \, dx = 0 \quad m = 0, 1, \ldots, p-1 \tag{2.19}$$

2.4 The Daubechies D4 Coefficients

In her landmark 1988 article [6], Ingrid Daubechies presented an entire family of wavelets which not only had all of the desirable conditions for wavelets, but also had compact support. Prior to 1988, other useful families of wavelets lacked either one of the conditions, or else lacked compact support.

In order to compare the Haar and Daubechies D4 wavelet, we first perform the following calculation. We begin with the Dilation Equation:

$$\phi(x) = \sum_n \alpha_n \, \phi(2x - n) \tag{2.20}$$

Taking its Fourier transform, we have

$$\hat{\phi}(\omega) = \sum_n \alpha_n \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \phi(2x - n) e^{-i\omega x} \, dx$$

$$= \frac{1}{2} \sum_n \alpha_n \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{-i(2x-n)\frac{\omega}{2}} \, e^{-in\frac{\omega}{2}} \, \phi(2x - n) \, d(2x - n)$$

$$= \frac{1}{2} (\sum_n \alpha_n e^{-in\frac{\omega}{2}}) \hat{\phi}(\frac{\omega}{2})$$

The expression

$$= \frac{1}{2} (\sum_n \alpha_n e^{-in\frac{\omega}{2}})$$

is variously written as $P(\frac{\omega}{2})$, the "symbol" of the dilation equation, or in [5] as $m_0(\frac{\xi}{2})$.

This means that $m_0(\xi) = \frac{1}{2} \sum_n \alpha_n e^{-in\xi}$. Daubechies uses $|m_0|$ in Chapter 6 of [5] as a useful yardstick for comparing the various members of the family of wavelets which she discovered. On page 31, we present a comparison of the Haar and the Daubechies D4 wavelets using this basis. The dotted line represents an ideal low—pass filter. The dashed line is $|m_0|$ calculated for the Haar low—pass filter, and the solid line is the same for the Daubechies D4 low—pass filter. It is readily apparent that the D4 filter is closer to the ideal. In particular, it is much flatter in the center and at the extremes, which earns the Daubechies filters the title of "maxflat" filters.

Strang [13] describes the Daubechies wavelets as follows: the lowpass filters have $p = 1, 2, 3, 4, \ldots$ zeros at π. They have $2p = 2, 4, 6, 8, \ldots$ coefficients,

and support on the interval $[0, N] = [0, 2p - 1]$. As p increases, the filters are increasingly "regular" and the wavelets are increasingly "smooth" (use of quotation marks is Strang's). The very first member of the family, with $p = 1$, is simply the Haar wavelet. Many of those in the field consider the Daubechies wavelets to begin with $p = 2$, which has $2p = 4$ coefficients, the Daubechies D4 wavelet.

Our experience with the application of wavelets to heartbeats has been that the Haar wavelet is highly unsatisfactory for this particular purpose. A "denoised" heartbeat using the Haar wavelet looks as though all of the peaks have been snipped off with a pair of scissors. Among the Daubechies family of wavelets, the higher values of p result in artifacts resembling sine waves being imposed on the denoised versions. Consequently, the wavelet of choice for this particular application is the Daubechies D4 wavelet.

We shall now present a brief derivation of the values of the four Daubechies D4 coefficients.

Condition 1

$$\alpha_0 + \alpha_1 + \alpha_2 + \alpha_3 = 2 \tag{2.21}$$

Condition 2 with m = 0

$$\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 2 \tag{2.22}$$

Condition 2 with m = 1

$$\alpha_3\alpha_1 + \alpha_0\alpha_2 = 0 \tag{2.23}$$

Condition 3 with m = 0

$$\alpha_0 - \alpha_1 + \alpha_2 - \alpha_3 = 0 \tag{2.24}$$

Condition 3 with m = 1

$$-\alpha_1 + 2\alpha_2 - 3\alpha_3 = 0 \tag{2.25}$$

Now we proceed in a straightforward algebraic way. Adding (2.21) and (2.24) together gives

$$2\alpha_0 + 2\alpha_2 = 2 \tag{2.26}$$

from which

$$\alpha_2 = 1 - \alpha_0 \tag{2.27}$$

Now take (2.21) and subtract (2.24) to obtain

$$2\alpha_1 + 2\alpha_3 = 2 \tag{2.28}$$

from which

$$\alpha_3 = 1 - \alpha_1 \tag{2.29}$$

Now recall (2.22)

$$\alpha_0^2 + \alpha_1^2 + \alpha_2^2 + \alpha_3^2 = 2 \tag{2.30}$$

$$\Rightarrow \alpha_0^2 + \alpha_1^2 + (1-\alpha_0)^2 + (1-\alpha_1)^2 = 2 \tag{2.31}$$

$$\Rightarrow 2\alpha_0^2 - 2\alpha_0 + 2\alpha_1^2 - 2\alpha_1 = 0 \tag{2.32}$$

$$\Rightarrow \alpha_0(\alpha_0 - 1) + \alpha_1(\alpha_1 - 1) = 0 \tag{2.33}$$

Note that this agrees with Condition 2 with m = 1, which means that equations (2.21) to (2.25) are dependent.

Now recall

$$-\alpha_1 + 2\alpha_2 - 3\alpha_3 = 0 \tag{2.34}$$

substituting into (2.34) from (2.27) and (2.29)

$$-\alpha_1 + 2(1 - \alpha_0) - 3(1 - \alpha_1) = 0 \tag{2.35}$$

$$\Rightarrow\ 2\alpha_1 - 2\alpha_0 = 1 \tag{2.36}$$

$$\Rightarrow\ \alpha_1 = \frac{1}{2} + \alpha_0 \tag{2.37}$$

combining (2.29) and (2.37)

$$\alpha_3 = 1 - \alpha_1 = 1 - (\frac{1}{2} + \alpha_0) = \frac{1}{2} - \alpha_0 \tag{2.38}$$

substituting into (2.23) from (2.27) and (2.29) and negating

$$\alpha_0(\alpha_0 - 1) + \alpha_1(\alpha_1 - 1) = 0 \tag{2.39}$$

substituting into (2.39) from (2.37)

$$\alpha_0(\alpha_0 - 1) + (\frac{1}{2} + \alpha_0)(-\frac{1}{2} + \alpha_0) = 0 \tag{2.40}$$

which has solution

$$\alpha_0 = \frac{1 \pm \sqrt{3}}{4} \tag{2.41}$$

from which we choose arbitrarily

$$\alpha_0 = \frac{1 + \sqrt{3}}{4} \tag{2.42}$$

then

$$\alpha_1 = \frac{1}{2} + \alpha_0 = \frac{3+\sqrt{3}}{4} \tag{2.43}$$

$$\alpha_2 = 1 - \alpha_0 = \frac{3-\sqrt{3}}{4} \tag{2.44}$$

$$\alpha_3 = \frac{1}{2} - \alpha_0 = \frac{1-\sqrt{3}}{4} \tag{2.45}$$

Condition 3 (A — Approximation) on the Daubechies D4 Coefficients

$$\sum_{n\in\mathbb{Z}} (-1)^n \, n^m \, \alpha_n \;=\; 0 \;\; for \; m = 0, 1, \ldots, p-1 \tag{2.46}$$

We have seen already that equality holds for $m = 0$ and $m = 1$, however for $m = 2$

$$\sum_{n\in\mathbb{Z}} (-1)^n \, n^2 \, \alpha_n \;= 0 - \alpha_1 + 4\alpha_2 - 9\alpha_3 = \sqrt{3} \;\neq\; 0 \tag{2.47}$$

from which we conclude that for the D4 Coefficients, $p = 2$. Consequently only the polynomials in 1 and x are linear combinations of the $\phi(x-n)$. In Daubechies' notation, D4 has $N = 2$; i.e., 2 vanishing moments.

The reader who wishes to learn more about the Continuous Wavelet Transform is referred to [5]. We shall make use of various Discrete Wavelet Transforms based on orthogonal matrices.

2.5 The Haar Discrete Wavelet Transform

For several decades before the 1980's, engineers and applied mathematicians in the field of signal analysis have been concerned with the construction of efficient low—pass and high—pass filters. These function like the bass and

treble controls on a household stereo system, allowing respectively low and high frequency sounds to pass. Many de—noising schemes are based on the notion that frequencies which are unusually higher than the rest of a signal are likely to be noise, and can safely be filtered out.

Consequently, long before there was such an entity as the Daubechies D4 coefficients, there was a ready—made framework for their use. Strang and Nguyen [13] refer to a digital filter as a combination of delays and advances of the signal. When it consists strictly of delays, and only a finite number of them, it is referred to as a "causal finite—impulse—response (FIR) filter". The coefficients of such a filter are designated by $h(n)$, which is the notation used by Daubechies for the family of coefficients which she discovered. This thesis will not dwell at length on the topic of signal processing; the interested reader is referred to [13] and [9].

Before proceeding further, we will construct a lowpass filter using the father wavelet, ϕ. Construction of the highpass filter using the mother wavelet is very similar and will not be presented here.

We assume a signal to be given by discrete (sampled) values at the finest scale possible; i.e., no finer scale information is available. For illustrative purposes, let us consider the finest scale to be V_0.

In our multiscale analysis, each of the subspaces V_j is spanned by $\phi_{j,\ell} = 2^{-\frac{j}{2}}\,\phi(2^{-j}x - \ell)$. Since $f \in V_0$,

$$f(x) = \sum_{n\in\mathbb{Z}} a_n^0\,\phi_{0,n} = \sum_{n\in\mathbb{Z}} a_n^0\,\phi(x-n)$$

We wish to construct a low-pass filter which is a projection $P_1 : V_0 \to V_1$. That is, a low-pass filter is a projection of the element of V_0 onto the next

coarser scale, V_1. Because $P_1\, f \in V_1$ it can be represented by

$$P_1\, f = \sum_{\ell \in \mathbf{Z}} a_\ell^1\, \phi_{1,\ell} = \sum_{\ell \in \mathbf{Z}} a_\ell^1\, \frac{1}{\sqrt{2}} \phi(\frac{x}{2} - n)$$

So we have

$$P_1\, f = \sum_{\ell \in \mathbf{Z}} a_\ell^1\, \phi_{1,\ell} = \sum_{k \in \mathbf{Z}} a_k^0\, P_1\, \phi_{0,k}$$

by the linearity of projection.

We assume that we know the a_k^0; what we need is a way to determine the a_ℓ^1.

We have

$$P_1\, \phi_{0,k} = \sum_{\ell \in \mathbf{Z}} < \phi_{1,\ell}, \phi_{0,k} > \phi_{1,\ell}$$

Now

$$< \phi_{1,\ell}, \phi_{0,k} >= \int \frac{1}{\sqrt{2}}\, \phi(\frac{x}{2} - \ell)\, \phi(x - k)\, dx \tag{2.48}$$

$$= \frac{1}{\sqrt{2}} \int \phi(\frac{x}{2} - \ell)\, \phi(x - k)\, dx \tag{2.49}$$

By the Dilation Equation

$$\phi(x) = \sum_{j \in \mathbf{Z}} \alpha_j\, \phi(2x - j) \tag{2.50}$$

so that

$$\phi(\frac{x}{2}) = \sum_{j\in\mathbf{Z}} \alpha_j \, \phi(x-j) \tag{2.51}$$

$$\phi(\frac{x}{2} - \ell) = \sum_{j\in\mathbf{Z}} \alpha_j \, \phi(x-2\ell-j) \tag{2.52}$$

Consequently,

$$< \phi_{1,\ell}, \phi_{0,k} >= \frac{1}{\sqrt{2}} \int \sum_{j\in\mathbf{Z}} \phi(x-2\ell-j) \, \phi(x-k) \, dx \tag{2.53}$$

But the ϕ are an orthonormal basis for V_0, so the only terms which appear are those for which $k = 2\ell + j$, i.e. only one term, so that

$$< \phi_{1,\ell}, \phi_{0,k} >= \frac{1}{\sqrt{2}} \alpha_{k-2\ell} \tag{2.54}$$

So now

$$P_1\phi_{0,k} = \sum_{\ell\in\mathbf{Z}} < \phi_{1,\ell}, \phi_{0,k} > \phi_{1,\ell} \tag{2.55}$$

$$= \sum_{\ell\in\mathbf{Z}} \frac{1}{\sqrt{2}} \alpha_{k-2\ell} \, \phi_{1,\ell} \tag{2.56}$$

and

$$P_1 f = \sum_{k\in\mathbf{Z}} a_k^0 \, P_1 \, \phi_{0,k} \tag{2.57}$$

$$= \sum_{k\in\mathbf{Z}} a_k^0 \sum_{\ell\in\mathbf{Z}} \frac{1}{\sqrt{2}} \, \alpha_{k-2\ell} \, \phi_{1,\ell} \tag{2.58}$$

$$= \frac{1}{\sqrt{2}} \sum_{\ell\in\mathbf{Z}} (\sum_{k\in\mathbf{Z}} \alpha_{k-2\ell} \, a_k^0 \,) \, \phi_{1,\ell} \tag{2.59}$$

but this is equal to

$$\sum_{\ell \in \mathbf{Z}} a_\ell^1 \, \phi_{1,\ell} \tag{2.60}$$

so that

$$a_\ell^1 = \frac{1}{\sqrt{2}} \sum_{k \in \mathbf{Z}} \alpha_{k-2\ell} \, a_k^0 \tag{2.61}$$

In a similar fashion, we obtain for the corresponding highpass filter

$$b_\ell^1 = \frac{1}{\sqrt{2}} \sum_{k \in \mathbf{Z}} (-1)^k \alpha_{2\ell-k+1} \, a_k^0 \tag{2.62}$$

The output of the highpass filter is the difference between the original signal and the output of the lowpass filter, and is orthogonal to the output of the lowpass filter.

Finally, having derived the lowpass and highpass filters, in practice we find it more convenient to operate on the sampled values of $f(x)$ *as though they were* the coefficients a_k^0, instead of the coefficients a_k^0 themselves. By doing this, results of de—noising and data compression algorithms are in terms of sampled values of $f(x)$.

Before considering the Daubechies D4 coefficients, we will examine the Haar coefficients, which can be considered the first members of the Daubechies family, the D2 coefficients.

We operate on a data vector whose length is an integer power of two. In most papers, [10], the format is as follows:

$y = y_i = (2, 3, 5, 7, 11, 13, 17, 19)$ (by way of example)

Recall that for the Haar wavelet, $\alpha_0 = \alpha_1 = 1$. Now by equation (2.61) and (2.62) we have

$$\frac{1}{\sqrt{2}} \begin{bmatrix} \alpha_0 & \alpha_1 & & & & & & \\ \alpha_1 & -\alpha_0 & & & & & & \\ & & \alpha_0 & \alpha_1 & & & & \\ & & \alpha_1 & -\alpha_0 & & & & \\ & & & & \alpha_0 & \alpha_1 & & \\ & & & & \alpha_1 & -\alpha_0 & & \\ & & & & & & \alpha_0 & \alpha_1 \\ & & & & & & \alpha_1 & -\alpha_0 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \\ 7 \\ 11 \\ 13 \\ 17 \\ 19 \end{bmatrix}$$

$$= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & & & & & & \\ 1 & -1 & & & & & & \\ & & 1 & 1 & & & & \\ & & 1 & -1 & & & & \\ & & & & 1 & 1 & & \\ & & & & 1 & -1 & & \\ & & & & & & 1 & 1 \\ & & & & & & 1 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 5 \\ 7 \\ 11 \\ 13 \\ 17 \\ 19 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{5}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ \frac{12}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ \frac{24}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ \frac{36}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix} := \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix}$$

We now permute:

$$\begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \longrightarrow \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} \frac{5}{\sqrt{2}} \\ \frac{12}{\sqrt{2}} \\ \frac{24}{\sqrt{2}} \\ \frac{36}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix}$$

Now we operate only on $s_1, \ldots, s_4$, the result of the low pass filter and leave $d_1, \ldots, d_4$, the result of the high pass filter.

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{5}{\sqrt{2}} \\ \frac{12}{\sqrt{2}} \\ \frac{24}{\sqrt{2}} \\ \frac{36}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{17}{2} \\ -\frac{7}{2} \\ 30 \\ -6 \end{bmatrix} = \begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \end{bmatrix}$$

and permute

$$\begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \end{bmatrix} \longrightarrow \begin{bmatrix} S_1 \\ S_2 \\ D_1 \\ D_2 \end{bmatrix} = \begin{bmatrix} \frac{17}{2} \\ 30 \\ -\frac{7}{2} \\ -6 \end{bmatrix}$$

In [10] the process stops when the first two terms only are S's, but that is because they are using the Daubechies D4 coefficients. With Haar coefficients, we can proceed until there is only one S.

$$\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{17}{2} \\ 30 \end{bmatrix} = \begin{bmatrix} \frac{77}{2\sqrt{2}} \\ -\frac{43}{2\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \mathbb{S}_1 \\ \mathbb{D}_1 \end{bmatrix}$$

The final result is:

$$\begin{bmatrix} \mathbb{S}_1 \\ \mathbb{D}_1 \\ D_1 \\ D_2 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} \frac{77}{2\sqrt{2}} \\ -\frac{43}{2\sqrt{2}} \\ -\frac{7}{2} \\ -6 \\ -\frac{1}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \\ -\frac{2}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 27.2236 \\ -15.2028 \\ -3.5000 \\ -6.0000 \\ -0.7071 \\ -1.4142 \\ -1.4142 \\ -1.4142 \end{bmatrix}$$

The sum of the squares of the entries in the original 8—vector is 1027. The sum of the squares of the four largest entries, 11, 13, 17, and 19, is 940. We say, following signal processing usage, that those four entries contain 91.53% of the "energy" of the signal. Note how in the final transformed result, the four components with the largest absolute value now contain 99.37% of the "energy". More of the energy of the signal is now carried in the same number of coefficients. As will be defined in Section 3.2, this means that the final transformed result has a lower *entropy functional* associated with it.

2.6 The Daubechies D4 Discrete Wavelet Transform

Again, we operate on a data vector whose length is an integer power of two. Again we use equations (2.61) and (2.62). Recall that:

$$\alpha_0 = \frac{1+\sqrt{3}}{4} \tag{2.63}$$

$$\alpha_1 = \frac{3+\sqrt{3}}{4} \tag{2.64}$$

$$\alpha_2 = \frac{3-\sqrt{3}}{4} \tag{2.65}$$

$$\alpha_3 = \frac{1-\sqrt{3}}{4} \tag{2.66}$$

Now

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & & & & \\ \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 & & all & zeros & \\ & & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 & & \\ & & \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 & & \\ & & & & \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ & all & zeros & & \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 \\ \alpha_2 & \alpha_3 & & & & & \alpha_0 & \alpha_1 \\ \alpha_1 & -\alpha_0 & & & & & \alpha_3 & -\alpha_2 \end{bmatrix}\begin{bmatrix} 2 \\ 3 \\ 5 \\ 7 \\ 11 \\ 13 \\ 17 \\ 19 \end{bmatrix} = \begin{bmatrix} 3.6903 \\ -0.1294 \\ 9.0537 \\ 0.7071 \\ 17.5390 \\ 0.7071 \\ 24.1642 \\ -6.2346 \end{bmatrix} = \begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix}$$

Now permute:

$$\begin{bmatrix} s_1 \\ d_1 \\ s_2 \\ d_2 \\ s_3 \\ d_3 \\ s_4 \\ d_4 \end{bmatrix} \longrightarrow \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} 3.6903 \\ 9.0537 \\ 17.5390 \\ 24.1642 \\ -0.1294 \\ 0.7071 \\ 0.7071 \\ -6.2346 \end{bmatrix}$$

As with the Haar coefficients, we now operate only on $s_1, \ldots, s_4$ and leave $d_1, \ldots, d_4$.

$$\frac{1}{\sqrt{2}}\begin{bmatrix} \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ \alpha_3 & -\alpha_2 & \alpha_1 & -\alpha_0 \\ \alpha_2 & \alpha_3 & \alpha_0 & \alpha_1 \\ \alpha_1 & -\alpha_0 & \alpha_3 & -\alpha_2 \end{bmatrix}\begin{bmatrix} 3.6903 \\ 9.0537 \\ 17.5390 \\ 24.1642 \end{bmatrix} = \begin{bmatrix} 10.1600 \\ 0.4943 \\ 28.3400 \\ -8.9716 \end{bmatrix} = \begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \end{bmatrix}$$

Now permute:

$$\begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \end{bmatrix} \longrightarrow \begin{bmatrix} S_1 \\ S_2 \\ D_1 \\ D_2 \end{bmatrix} = \begin{bmatrix} 10.1600 \\ 28.3400 \\ 0.4943 \\ -8.9716 \end{bmatrix}$$

We cannot proceed further since we have only S_1 andS_2 to process, and D4 requires a vector of length at least 4 in order to operate.

The final result is:

$$\begin{bmatrix} S_1 \\ S_2 \\ D_1 \\ D_2 \\ d_1 \\ d_2 \\ d_3 \\ d_4 \end{bmatrix} = \begin{bmatrix} 10.1600 \\ 28.3400 \\ 0.4943 \\ -8.9716 \\ -0.1294 \\ 0.7071 \\ 0.7071 \\ -6.2346 \end{bmatrix}$$

In the Haar example, the four largest coefficients in absolute value went from 91.53% of the "energy" of the signal to 99.37% after the transformation. Now, in the Daubechies D4 example, they contain an even higher 99.87% of the energy, resulting in still lower entropy functionals associated with the transformed result.

These results are entirely consistent with our goals of achieving a higher concentration of the "energy" of the signal in fewer coefficients in the wavelet transform.

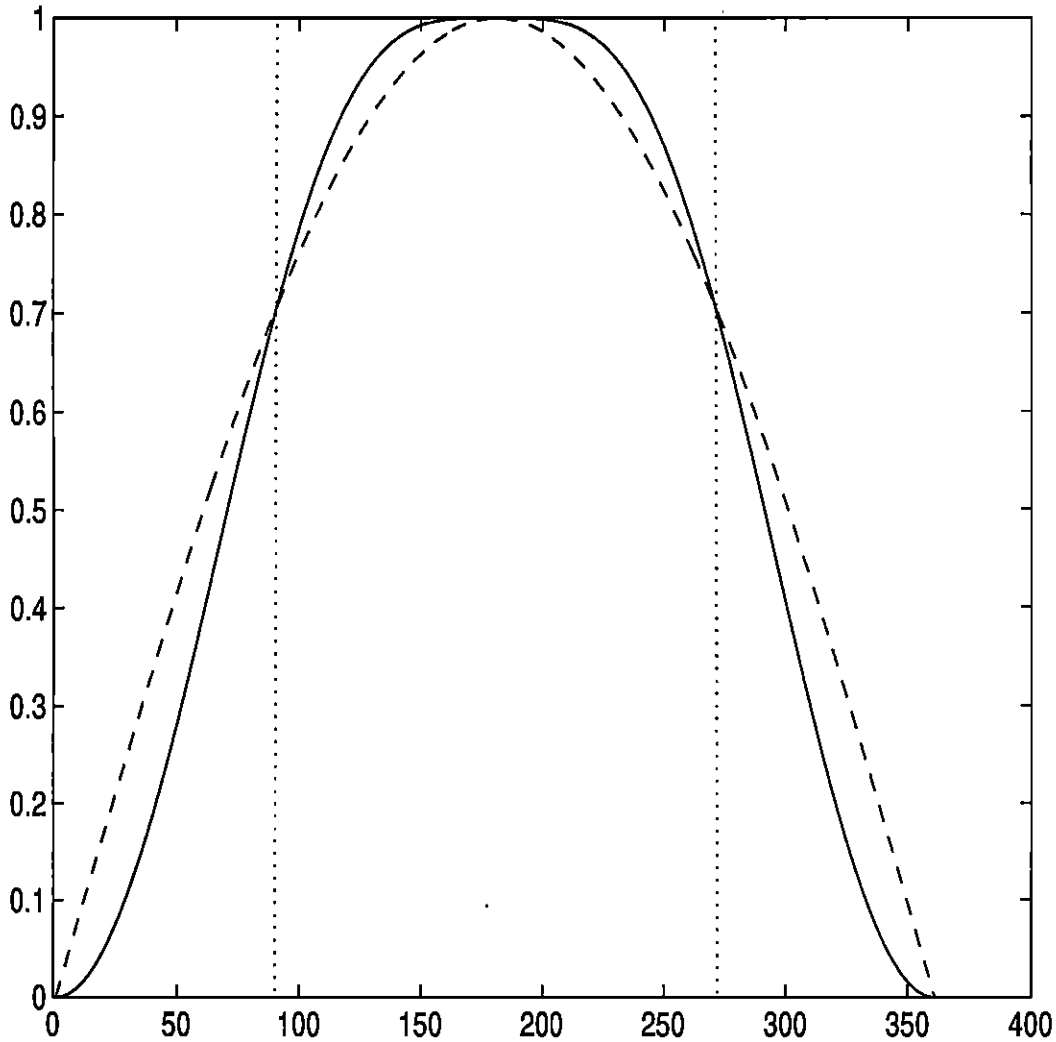


Figure 2.1: Comparison of $|m_0|$ for Haar and Daubechies D4 low—pass filters *The dotted line is an ideal low—pass filter. The dashed line is the Haar low—pass filter, and the solid line is the Daubechies D4 low—pass filter. The values on the x—axis would normally run from $-\pi$ to π but have been rescaled to 1 to 361 for maximum clarity.*

Chapter 3

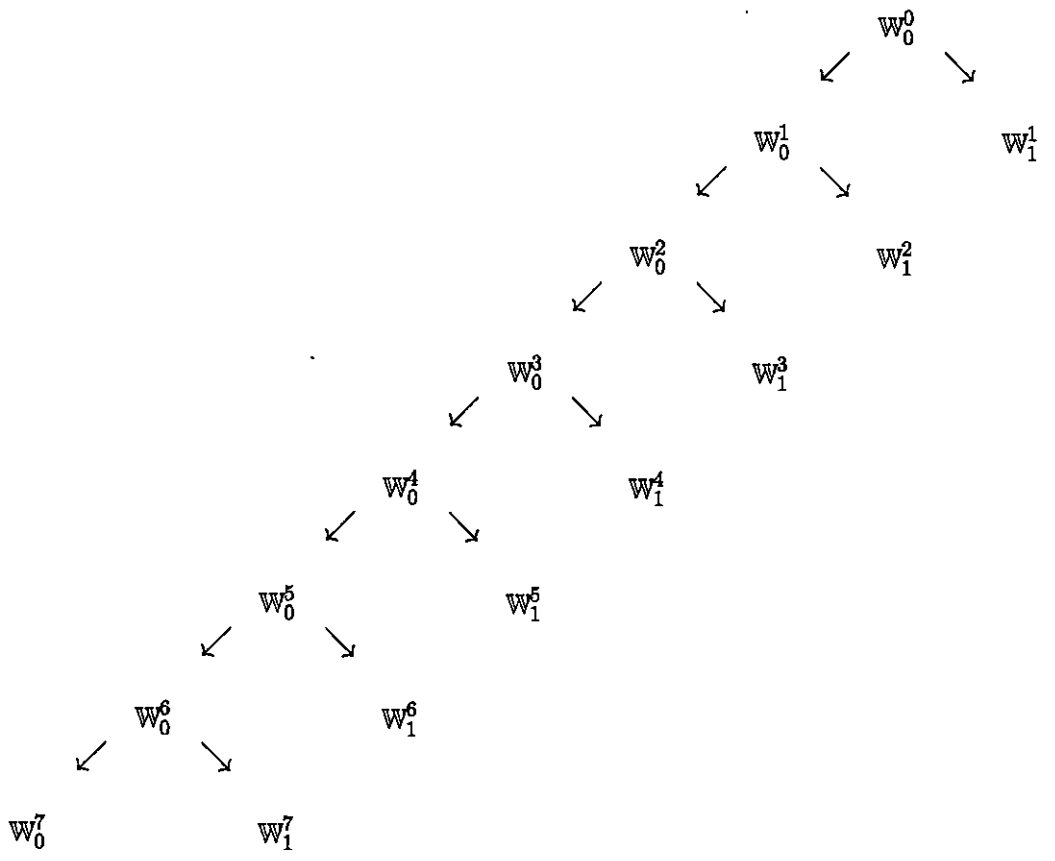
Wavelet Analysis of Electrocardiograms

3.1 Preliminaries

For testing purposes, twenty heartbeats, each of approximately one second's duration, were chosen from a library of recordings obtained from a local medical software company. The company is developing a portable arrhythmia detector which can be worn comfortably by ambulatory outpatients for twenty—four hours. The data is stored on a $3\frac{1}{2}$ inch diskette, which is then analyzed using a 486 or Pentium—based personal computer.

The heartbeats are sampled at the rate of 250 samples/second, so for an average beat with a duration of approximately one second, a 256—vector is used.

Applying the pyramid algorithm, we obtain:



At each node, the arrow pointing left and down points to the node containing the result of the low—pass filter, and the one pointing right and down to that containing the result of the high—pass filter.

The following definition of "binary tree" is due to [4].

Definition 3.1 *Binary trees are best defined recursively. A binary tree T is a structure defined on a finite set of nodes that either contains no nodes, or is comprised of three disjoint sets of nodes: a root node, a binary tree called its left subtree, and a binary tree called its right subtree. The binary tree that*

contains no nodes is called the empty tree or the null tree, sometimes denoted NIL. *If the left subtree is nonempty, its root is called the left child of the root of the entire tree. Likewise, the root of a nonnull right subtree is the right child of the root of the entire tree. If a subtree is the null tree* NIL, *we say that the child is absent or missing. In this thesis, we will only be concerned with binary trees in which each node has either a left child and a right child, or else has no children. Such trees are usually denoted as full binary trees.*

The question naturally arose, since we are analyzing heartbeats, are there other choices of nodes in the binary tree which would be better in some sense? The first requirement was a precise definition of "better". After a heartbeat has been subjected to a wavelet transform, we next want to perform denoising of the signal, and then arrhythmia detection. To provide the best precondition for these processes, we consider one wavelet transform to be superior to another if it contains more of the power of the signal in fewer coefficients.

The following discussion of measures of information and the entropy criterion and estimates is from [2].

Definition 3.2 *An additive measure of information on* ℓ^2 *is any functional* $\mathbb{M} : \ell^2 \to \mathbb{R}^+$ *satisfying*

$$(i)\ \mathbb{M}(x \times y) = \mathbb{M}(x) + \mathbb{M}(y)\ \textit{for every pair of sequences}\ x, y, \tag{3.1}$$

$$(ii)\ \mathbb{M}(0) = 0. \tag{3.2}$$

Here $x \times y \in \ell^2 \times \ell^2 \cong \ell^2$; *an example of a suitable isomorphism is*

$$(x_1, x_2, \ldots) \times (y_1, y_2, \ldots) \longrightarrow (x_1, y_1, x_2, y_2, \ldots) \in \ell^2 \tag{3.3}$$

3.2 Entropy Functionals

Let $x \in \ell^2$ and denote by $\|x\|$ the usual norm: $\|x\|^2 = \sum_k |x|^2$. Then the sequence defined by $|x|^2/\|x\|^2$ gives a probability distribution of the "energy" of x.

Definition 3.3 *The Shannon entropy of this distribution is*

$$\mathbb{H}(x) = -\sum_k (|x_k|^2/\|x\|^2) \log(|x_k|^2/\|x\|^2) \tag{3.4}$$

where the summand is interpreted by convention as 0 for any $x_k = 0$, since $\lim_{h\to 0} h \log h = 0$.

The Shannon entropy is a well-known measure of the expected information content of a distribution.

Definition 3.4 *The $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional is defined as*

$$H(x) = -\sum_k |x_k|^2 \log |x_k|^2 \tag{3.5}$$

with the same convention for the case $x_k = 0$.

Note that

$$\mathbb{H}(x) = -\sum_k (|x_k|^2/\|x\|^2) \log(|x_k|^2/\|x\|^2) \tag{3.6}$$

$$= -\frac{1}{\|x\|^2} \sum_k |x_k|^2 (\log |x_k|^2 - \log \|x\|^2) \tag{3.7}$$

$$= -\frac{1}{\|x\|^2} \sum_k |x_k|^2 \log |x_k|^2 + \sum_k (|x_k|^2/\|x\|^2) \log \|x\|^2 \tag{3.8}$$

$$= H(x)\|x\|^{-2} + \log \|x\|^2 \tag{3.9}$$

By (3.10—3.13), H is obviously an additive measure of information; $\mathbb{H}$ is not.

$$H(x \times y) = H(x_1, y_1, x_2, y_2, \ldots) \tag{3.10}$$

$$= -\sum_k (|x_k|^2 \log |x_k|^2 + |y_k|^2 \log |y_k|^2) \tag{3.11}$$

$$= -\sum_k |x_k|^2 \log |x_k|^2 - \sum_k (|y_k|^2 \log |y_k|^2 \tag{3.12}$$

$$= H(x) + H(y). \tag{3.13}$$

and $H(0) = 0$ by our convention that the summand is interpreted as 0 for any $x_k \doteq 0$. It is not necessary to prove that $\mathbb{H}$ is not an additive measure of information on ℓ^2 since we will see numerous counterexamples in the following pages. We do have, however, that whenever $\|x\| = \|y\|$, we have $\mathbb{H}(x) < \mathbb{H}(y) \Leftrightarrow H(x) < H(y)$.

As an additive measure of information, we will use H in a technique known as a best—basis algorithm. As a non—additive measure of information, the use of $\mathbb{H}$ will be restricted to a technique known as a near—best—basis algorithm.

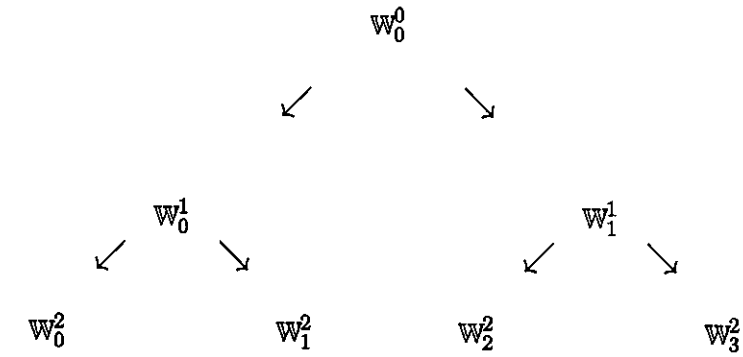
The objective is to compare the discrete wavelet transforms resulting from the Pyramid algorithm with those using a near—best—basis algorithm and then a best—basis algorithm. We shall be particularly interested in how significant the improvement is, and whether the improvement is sufficient to justify the increased cost of computation.

3.3 Best Basis Selection

Coifman and Wickerhauser [2], [3] deal primarily with the process of best—basis selection in the sense of an additive measure of information such as H,

the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional. It suggests near—best—basis selection in the sense of a non—additive measure of information such as $\mathbb{H}$, the Shannon entropy. The topic of near—best—basis selection is covered thoroughly in Taswell [15]. The following discussion is modified slightly to real vectors of length $N = 2^n$, where $n \in \mathbb{Z}^+$.

Let $\mathbb{W}$ be a binary tree of successive low and high—pass filterings of the data vector $y = (y_1, \ldots, y_{2^n})$. Here, for clarity, we show only the top three levels of the tree which has eight levels, for a total of 255 nodes.



We now describe the best—basis search algorithm. What is meant by best—basis is that the algorithm will search the finite binary tree $\mathbb{W}$ and find a finite collection of nodes whose direct sum is ℓ^2, such that the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional H of the discrete wavelet transform of the data vector y is less than or equal to that resulting from any other possible finite collection of nodes whose direct sum is also ℓ^2.

Mark all nodes at the greatest depth (i.e. maximal nodes) of the binary tree $\mathbb{W}$ as "kept", by setting an indicator for that node at "1". All other nodes have their indicator set to "0". Using the Haar DWT, the superscript of the nodes at this level will be n, where y is of length 2^n. With the Daubechies D4

DWT, the maximal nodes are 2—vectors, and they have superscript $n-1$.

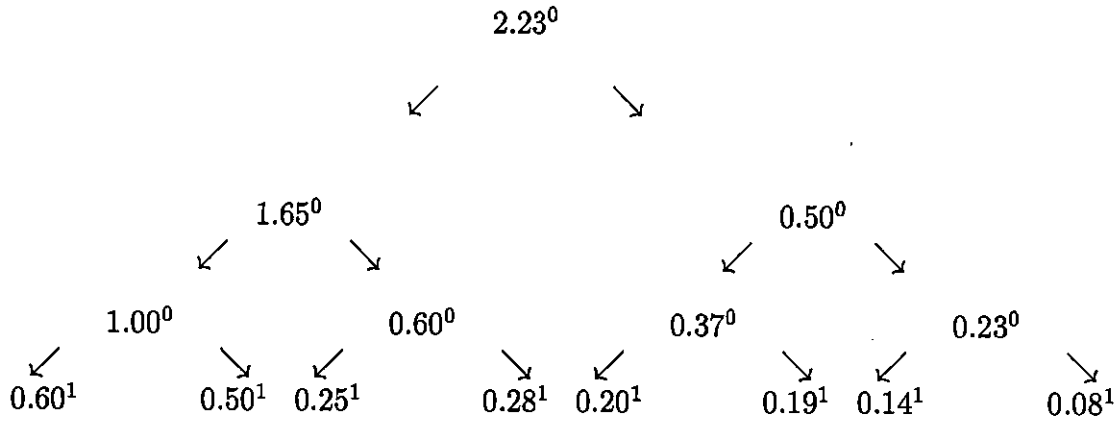
Now the search begins at the node $\mathbb{W}_0^{n-1}$ (Haar) or $\mathbb{W}_0^{n-2}$ (Daubechies D4), that is, the leftmost node in the next level above the maximal nodes. We will use the Daubechies D4 tree as an example.

We now introduce a convention which will permit us to make meaningful comparisons between any node and all of the "kept" nodes in its subtree. Using the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional, H, which is an additive cost measure, we introduce $M(\mathbb{W}_p^q)$, where $q = 0, \ldots, (n-1)$, then $p = 0, \ldots, 2^q$. Initially, $M(\mathbb{W}_p^q)$ is set equal to $H(\mathbb{W}_p^q)$. For a given node, $H(\mathbb{W}_p^q)$ will be fixed, but $M(\mathbb{W}_p^q)$ may change in value. We begin by comparing $M(\mathbb{W}_0^{n-2})$ with $M(\mathbb{W}_0^{n-1})$ + $M(\mathbb{W}_1^{n-1})$, the sum for its two children.

If $M(\mathbb{W}_0^{n-2}) \leq M(\mathbb{W}_0^{n-1}) + M(\mathbb{W}_1^{n-1})$, then we mark $\mathbb{W}_0^{n-2}$ as "kept" by changing its indicator from "0" to "1". We change the indicators for $\mathbb{W}_0^{n-1}$ and $\mathbb{W}_1^{n-1}$ to "not kept" or "0". In this case, the value of $M(\mathbb{W}_0^{n-2})$ does not change. However, if $M(\mathbb{W}_0^{n-2}) > M(\mathbb{W}_0^{n-1}) + M(\mathbb{W}_1^{n-1})$, then we reset $M(\mathbb{W}_0^{n-2})$ to equal $M(\mathbb{W}_0^{n-1}) + M(\mathbb{W}_1^{n-1})$, and leave the "kept" and "not kept" indicators unchanged. This later permits meaningful comparisons between any node and all of the nodes in its subtree which have been marked as "kept".

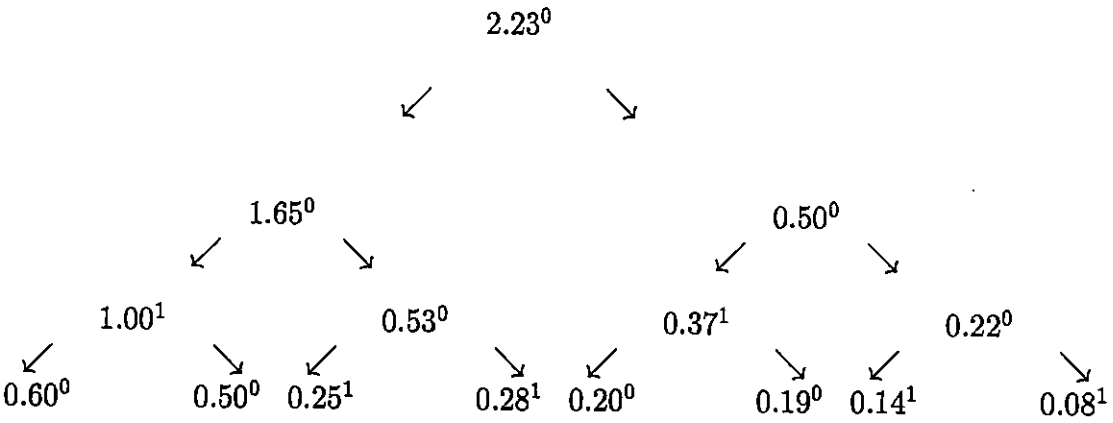
The process is now repeated at each successive node on the $\mathbb{W}_p^{n-2}$ level, for $p = 1, \ldots, s^{n-2}$. We then move up one level to $\mathbb{W}_0^{n-3}$ and across the $(n-3)$ level. Continuing up the tree in this manner, the final comparison is between $y = M(\mathbb{W}_0^0)$ and $M(\mathbb{W}_0^1) + M(\mathbb{W}_1^1)$.

Let us assume the following values for $M(\mathbb{W}_p^q)$ and observe the action of the best—basis selection. The superscripted zeros and ones are the "kept" indicators for the nodes.



On the first pass, across the $\mathbb{W}^2$ level, we accept $\mathbb{W}_0^2$, where $M(\mathbb{W}_0^2) = 1.00$. We turn its "kept" indicator to "1" and all those in its subtree to "0". $\mathbb{W}_1^2$, where $M(\mathbb{W}_1^2) = 0.60$ is not kept, so "kept" indicators do not change, but $M(\mathbb{W}_1^2)$ is adjusted to $M(\mathbb{W}_2^3) + M(\mathbb{W}_3^3) = 0.25 + 0.28 = 0.53$. $\mathbb{W}_2^2$ is kept and its indicator changed to "1" and those of its subtree to "0". $\mathbb{W}_3^2$, where $M(\mathbb{W}_3^2) = 0.23$ is not kept, and $M(\mathbb{W}_3^2)$ is reset to $M(\mathbb{W}_6^3) + M(\mathbb{W}_7^3) = 0.14 + 0.08 = 0.22$.

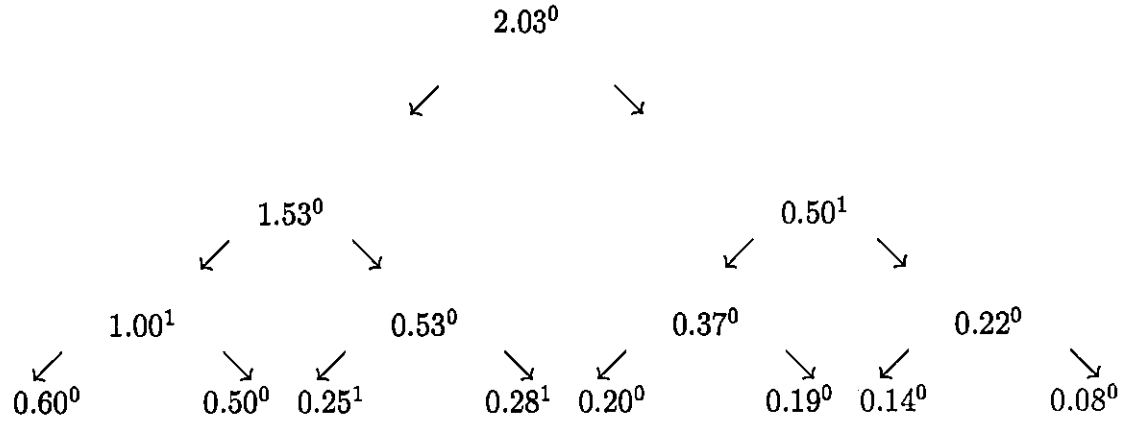
So now we have



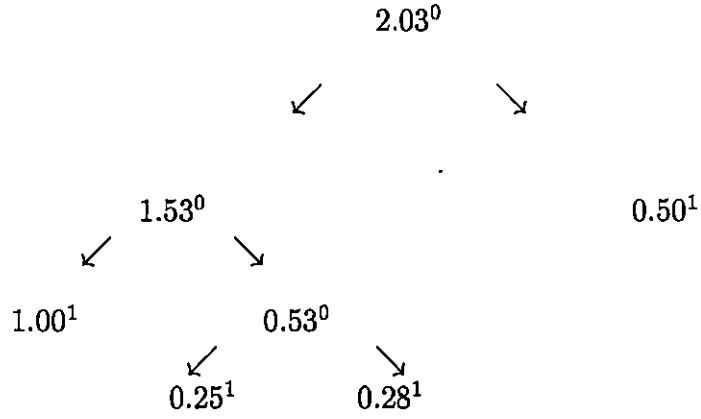
On the second pass, across the $\mathbb{W}^1$ level, $\mathbb{W}_0^1$ is not kept, so $M(\mathbb{W}_0^1)$ is reset to $M(\mathbb{W}_0^2) + M(\mathbb{W}_1^2) = 1.00 + 0.53 = 1.53$. $\mathbb{W}_1^1$ is kept, so its indicator changes to "1" and all those in its subtree to "0".

Lastly, $\mathbb{W}_0^0$ is not kept, and $M(\mathbb{W}_0^0)$ is reset to $M(\mathbb{W}_0^1) + M(\mathbb{W}_1^1) = 1.53 + 0.50 = 2.03$. This is the total information content of the subtree consisting of all nodes marked as "kept".

The final result is



That is, we are using the following "pruned" tree:



It is a Proposition of Coifman and Wickerhauser [2] that this process yields the best—basis representation of y, in that there does not exist another choice of nodes in the tree, from which we could reconstruct y, having lower $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional.

Proof. [2] Every minimal "kept" node W is the root of a subtree containing some of the maximal nodes of $\mathbb{W}$; i.e., some of the nodes farthest from the root of $\mathbb{W}$. This set of subtrees is disjoint, since if two subtrees intersect, then one must contain the other and so their roots cannot both be minimal. The union of these disjoint subtrees covers all of the maximal nodes of $\mathbb{W}$, which form a complete level of the tree.

So call this graph G. Notice that the sum $\sum_{\mathbb{W}_p^q \in G} M(\mathbb{W}_p^q)$ over the minimal "kept" nodes $W \in G$ is equal to $M(\mathbb{W}_0^0)$, where $\mathbb{W}_p^q$ is the root of $\mathbb{W}$. By our above—described search algorithm, this is the minimum achievable measure of information.

One significant feature of the best—basis selection is illustrated by a comparison of the number of possible subtrees with the number of comparisons which must be made.

For a tree with one level below the root, there are two possibilities. We either take the root as "kept", or both children.

For a tree with two levels below the root, there are two possibilities for each of the two subtrees below the root, or we can take the root as "kept". This yields $(2 \times 2) + 1 = 5$ choices.

For a tree with three levels below the root, there are 5 possibilities for each of the two subtrees below the root, or we can take the root as "kept". This yields $(5 \times 5) + 1 = 26$ choices.

We see that with this recursive definition, the number of possible subtrees is much greater than $2^{(2^{(n-2)})}$, where y is a vector of length 2^n, for the Daubechies D4 DWT, and greater than $2^{(2^{(n-1)})}$ for the Haar DWT.

In this case, with $N = 256 = 2^8$, we have that there are well over 1.8×10^{19} possible subtrees using the Daubechies D4 DWT, and well over 3.4×10^{38} possible subtrees using the Haar DWT. Fortunately, the best—basis search runs in $O(N)$ time.

3.4 Near—Best Basis Selection

The idea of the Near—Best Basis Selection is to start at the root of the binary tree, which is at the top, and search down. Using the Shannon entropy, $\mathbb{H}$, which is a non-additive measure of information content, we compare $\mathbb{H}$ at each node with the Shannon entropy of its two children. We do not renormalize the children; we treat the parent 2^k—vector as one vector and the two 2^{k-1}—vector children as a single 2^k—vector.

If the parent has lower Shannon Entropy, $\mathbb{H}$, than its two children, we accept the parent and do not accept any of the descendants of the parent node, even though there may be a better basis by doing so. It is quite easy to construct trees where the near—best basis selection will not agree with the best—basis selection. We are trying to achieve a result which is close to the best—basis result, but at a much lower computational cost. As we shall see in Section 5.1, although the Pyramid algorithm, the Near—Best—Basis and the Best Basis are all $O(N)$ algorithms, the constant associated with the first two is considerably lower than that for the last one.

If the two children have lower Shannon entropy, then we accept them. We now consider them as new parent nodes, and perform the same comparison of each of them with its two children. In computing the Shannon entropy, $\mathbb{H}$, of each new parent/children set, we renormalize to the length of the parent vector, so that comparisons with data from ancestral nodes is not meaningful.

It is entirely possible that our near—best—basis may contain only a few nodes. The question is whether we have achieved a significant improvement over the pyramid algorithm, as measured by the reduction in Shannon entropy, at a low additional computational cost.

Chapter 4

The Selection Processes

There are twenty samples in all. We will examine the processes at work on the first sample in minute detail. Having done so, for the remaining nineteen we will present only the final results for each sample, together with conclusions as to the effectiveness and value of the procedures.

The first sample is from a file code—named "c27wv97.asc", which contains a heartbeat sampled at 250 samples per second. The data has been discretized to multiples of 16, with a range of values from -400 to 400 to quantize the voltage recorded. The entire recording contains 30,136 data points,which is approximately two minutes. The first sample consists of $y(2350 : 2605)$, a 256-vector of just over one second's duration containing one complete heartbeat.

Note that the recording is noisy. There are three main sources of noise. The first source is caused by the discretization of the data. The other two sources are the muscles surrounding the heart, and the muscles of the heart itself (not directly related to the pumping of blood). One of the prime motivations for finding wavelet transforms of these signals with as much of the power of the signal contained in as few coefficients as practical is that any denoising

technique will perform better with such preconditioning.

The ambulatory recording device measures voltage difference between two electrodes positioned on the right and left sides of the patient's chest. It produces a tracing which is very similar to that of the Lead I of a 12—lead ECG machine as used in a hospital. Lead I has electrodes attached to the left and right wrists of the patient [12].

The MATLAB commands which were used to create the illustration for Sample 1 were:

```
fd = fopen('c27wv97.asc');
s=fscanf(fd,'%f');
fclose(fd);
size(s) output:  30136 × 1
sc=s(2350:2605);
plot(sc)
xlabel(time)
ylabel(voltage measure)
```

We begin by normalizing the vector. Otherwise, we will sometimes be dealing with positive logarithms and sometimes with negative, which although perfectly correct would nevertheless be confusing.

```
ssc = sc/norm(sc);
```

First we produce the DWT of Sample 1 using the Daub4 and Haar DWT's. Of the functions such as wavel, invwavel, wavelhaar, invwavelhaar, wavelent, wavelenthaar, invwavelent, invwavelenttrunc, wavelmod and wavelentnorm, only the first two were not written by the author.

```
ssa = wavel(ssc, daub4);
```

```
ssb = wavelhaar(ssc, haar2);
```

Now we proceed with the Near—Best—Basis selection using Daubechies D4 and Shannon entropy, $\mathbb{H}$.

```
ssd = wavelent(ssc,daub4);
```

This command takes the data vector `ssc`, which is the normalized version of the original data vector `sc`, and performs a low—pass and a high—pass filtering using the Daubechies D4 coefficients. The results of the low—pass filtering are stored in coefficients `ssd(1:128)`, while those for the high—pass filtering reside in `ssd(129:256)`. The Shannon entropy of the parent, `ssc`, is stored at `ssd(257)`, while that of the two children is stored at `ssd(258)`. We find that `ssd(257) = 3.0997`, while `ssd(258) = 2.4926`. Consequently, we accept the children.

```
se = ssd(1:128);
sse = wavelent(se, daub4);
sf = ssd(129:256);
ssf = wavelent(sf, daub4);
```

For `se`, we obtain a Shannon entropy of 2.3633 for the parent and 1.9698 for its children, and so we accept the children.

For `sf`, however, the parent has entropy of 3.3252, but the children have entropy of 3.6085, so we accept `sf`, and reject its children and all their possible descendants. Now we need only concern ourselves with the descendants of `se`.

```
sg = sse(1:64);
ssg = wavelent(sg, daub4);
sh = sse(65:128);
ssh = wavelent(sh, daub4);
```

For both sg and sh, we obtain the result that the children have lower entropy than the parent, 1.7920 vs. 1.6946 and 1.6848 vs. 1.1091 respectively. We accept the children.

```
si = ssg(1:32);
ssi = wavelent(si, daub4);
sj = ssg(33:64);
ssj = wavelent(sj, daub4);
```

This time, both sets of children have higher entropy than their parent, 1.2823 vs. 1.5248 and 0.8211 vs. 1.3989 respectively. We accept si and sj and reject their children and all possible descendants of their children.

```
sk = ssh(1:32);
ssk = wavelent(sk, daub4);
sl = ssh(33:64);
ssl = wavelent(sl, daub4);
```

We now obtain the interesting result that the children of sk have lower entropy, while those of sl have higher entropy, 3.0136 vs. 2.8205 and 0.5226 vs. 1.3008 respectively. We accept the children of sk, and we accept sl.

```
sm = ssk(1:16);
ssm = wavelent(sm, daub4);
sn = ssk(17:32);
ssn = wavelent(sn, daub4);
```

This time, the children of sm have higher entropy while those of sn have lower entropy. We accept sm, and the children of sn.

```
so = ssn(1:8);
sso = wavelent(so, daub4);
```

```
sp = ssn(9:16);
ssp = wavelent(sp, daub4);
```

Again we have differing results, with the children of so having lower, and the children of sp having higher entropy, 1.4770 vs. 1.4310 and 1.5709 vs. 1.6558 respectively. We accept the children of so, and stop at sp.

```
sq = sso(1:4);
ssq = wavelent(sso, daub4);
sr = sso(5:8);
ssr = wavelent(sr, daub4);
```

The children of sq have higher entropy, while those of sr have lower, 0.7153 vs. 0.9356 and 1.0464 vs. 0.6993 respectively. We stop at sq, and accept the children of sr.

```
ss = ssr(1:2);
st = ssr(3:4);
```

Of course, ss and st are only 2—vectors; they are too short to undergo a wavelet transform using Daub4 coefficients. Accordingly, we accept ss and st.

We now construct our Near—Best—Basis DWT of Sample 1 using the Daubechies D4 coefficients and Shannon entropy.

```
nearbest = [ si sj sm sq ss st sp sl sf ];
```

The Shannon entropy of the 256—vector "nearbest" is 1.9901. This is a reduction of 8.79% from the 2.1819 Shannon entropy of ssa, the DWT of Sample 1 using the Daubechies D4 coefficients and the Pyramid algorithm.

Now we consider the Best Basis for Sample 1. Since the binary tree which

must be checked contains 255 nodes, it is not practical to present a detailed treatment such as the one above for the Near—Best—Basis. The data vector is processed by a subroutine titled "best", which like the other subroutines is written in C language to run in a Matlab environment. The output of "best" is a vector of length 2303. Each level of the binary tree requires 256 positions, so eight levels require 2048. The remaining positions, 2049 to 2303, contain the $\mathbb{L}^2 log \mathbb{L}^2$ entropies of each of the 255 nodes of the binary tree. There is one exception to this rule, and that is that in the lowest level, it would be a waste of time to compute separately the entropies for both children, only to add them together for the purpose of comparison with the entropy of their parent. At the lowest level, therefore, each even—numbered node contains the total entropy of itself and its sibling, while the sibling node contains the value zero.

At this point, a new subroutine "pyr" was written to calculate the Shannon entropy of the wavelet transform of the original signal which results from the pyramid algorithm. Since the original signal has been normalized, its Shannon entropy, $\mathbb{H}$, is identical to its $\mathbb{L}^2 log \mathbb{L}^2$ entropy functional. Accordingly, unless it is necessary to do otherwise, we shall no longer distinguish between the two entropies, and just use the term "entropy" or "entropy functional" for the rest of the thesis. Rather than use the subroutine "wavel" and calculate the entropy of its output, we can now simply sum the entropies at nodes 3, 5, 9, 17, 33, 65, 128 and 129.

The objective now became a single command, "treat", which would call on all the existing subroutines. Its output would be the entropy of the pyramid algorithm, the entropy of the Near—Best—Basis and the "kept" nodes of its

binary tree, and the entropy of the Best Basis and the "kept" nodes of *its* binary tree.

After the experience of producing the Near—Best—Basis for Sample 1, it was apparent that a much more efficient method would be required to perform the Best Basis. This need to calculate the Best Basis efficiently resulted in a streamlining of the method of producing the Near—Best—Basis, which we now explain.

As was pointed out after Definition 2.3, whenever $\|x\| = \|y\|$, we have $\mathbb{H}(x) < \mathbb{H}(y) \Leftrightarrow H(x) < H(y)$. Now, a node and its two children have the same norm. This means that instead of performing the Near—Best—Basis algorithm as we did for Sample 1, we can use the additive $\mathbb{L}^2 log \mathbb{L}^2$ entropy instead of the non—additive Shannon entropy. Every choice between parent and children will have the same result as before, and we are relieved of the necessity of recalculating entropies for the Near—Best—Basis once we have them for the Best Basis.

This new procedure dramatically highlighted a global flaw of the non—additive Shannon entropy. Because it renorms at each step, nodes which contain virtually none of the power of the signal are considered at the same weight as nodes containing much of its power. This makes comparisons from one part of the tree to another entirely meaningless. The use of the $\mathbb{L}^2 log \mathbb{L}^2$ entropy functional for both searches completely eliminates this problem.

A subroutine "nb" operates on the output of "best" to produce the binary tree of "kept" and "not kept" nodes for the Near—Best—Basis. Once we have chosen a parent over its two children, all the descendants of the children are ineligible for further consideration. "nb" accomplishes this by setting their

indicators to -1. The resulting entropy is then calculated from the "kept" nodes, by the subroutine "nbent".

Calculation of the "kept" and "not kept" nodes for the Best Basis is made by subroutine "indic". During "indic", the entropy percolates to the root. Rather than calculate the entropy from the entropies of the "kept" nodes, it is simply read off from the root node of the output of "indic". "indic" outputs a 510—vector. Entries 1 to 255 are the M—measures mentioned at the beginning of Section 2.2. Entries 256 to 510 are "1" for a "kept" node and "0" for a "not kept" node.

A single subroutine "srt" reads the locations of the "kept" nodes for both the Near—Best—Basis and the Best Basis, and outputs them. A familiar convention from computer science has been adopted. The root node of the binary tree is labelled node 1. Its two children are then 2 and 3. Their children are 4 and 5, and 6 and 7 respectively. The children of node n are $2n$ and $2n+1$. This permits fast search algorithms to be written.

Now that "treat" was finished, it only remained to write a reconstruction algorithm, "recon", which would take any wavelet and any binary tree of "kept" and "not kept" nodes, and perform an inverse discrete wavelet transform. The two original programs, "wavel" and "invwavel" could only perform on the subtree associated with the Pyramid algorithm. As was pointed out earlier, in Chapter 3, this is only one of over 1.8×10^{19} possible subtrees. What is remarkable about "recon" is that it can reconstruct a signal which has been modified from its original wavelet representation by *any* denoising technique based on *any* of the possible subtrees. We are now free to carry out the usual treatments of deleting small coefficients and higher scales on the wavelets

which are output by both the Near—Best—Basis and the Best Basis. Detailed comparison of the differences in the reconstructed signals, however, is outside the scope of this thesis.

4.1 The Results

Results are presented in the following format:

Table 4.1: Results for 20 Samples

Sample Number	File Identification	Location in File
Sample 1	c27wv97.asc	2350:2605
Treatment	Wavelet Entropy	Nodes Kept
pyramid	2.1819	
Near—Best	1.9901	3 8 9 11 20 43 84 170 171
Best	1.9896	3 8 9 11 20 84 170 171 172 173 174 175

Sample 2	c27wv97.asc	11910:12165
pyramid	1.9148	
Near—Best	1.7888	3 5 8 9
Best	1.7872	5 6 8 9 61 62 113 114 115 116 117 119 121 126 127 224 225 236 237 240 241

Sample 3	c27wv97.asc	18097:18352
pyramid	1.9800	
Near—Best	1.8161	3 8 9 11 20 42 43
Best	1.8156	3 8 9 11 41 42 81 87 160 161 172 173

Sample 4	c27wv97.asc	28221:28476
pyramid	2.1881	
Near—Best	1.9692	3 4 10 11
Best	1.9692	3 4 10 11

Sample 5	c27wv54.asc	388:643
		(last 10 positions set to 0 as they are part of next beat)
pyramid	2.1795	
Near—Best	1.9323415	3 8 9 11 41 43 160 161 162 163 168 169 170 171
Best	1.9323355	3 8 9 11 41 160 161 162 163 168 169 170 171 172 173 174 175

Sample 6	c27wv54.asc	25134:25389
		(last 8 positions set to 0 as they are part of next beat)
pyramid	2.1327	
Near—Best	1.8543	3 8 9 10 11
Best	1.8541	3 8 9 11 42 43 82 160 161 162 163 166 167

Sample 7	c27wv54.asc	18760:19011
		(last 9 positions set to 0 as they are part of next beat)
pyramid	2.1189	
Near—Best	1.8684	3 8 9 10 11
Best	1.8684	3 8 9 10 11

Sample 8	c27wv54.asc	12761:13016
		(last 8 positions set to 0 as they are part of next beat)
pyramid	2.1297	
Near—Best	1.8992	3 8 9 10 11
Best	1.8992	3 8 9 10 11

Sample 9	c27wvd6.asc	26933:27188
pyramid	1.8498	
Near—Best	1.5563	3 5 8 9
Best	1.5563	3 5 8 9

Sample 10	c27wvd6.asc	27208:27463
		(note: samples 9 and 10 are neighbours)
pyramid	1.8706	
Near—Best	1.3868	3 5 8 9
Best	1.3868	3 5 8 9

Sample 11	c27wvd6.asc	8611:8866
pyramid	1.9110	
Near—Best	1.6159	3 8 9 10 11
Best	1.6159	3 8 9 10 11

Sample 12	c27wvd6.asc	3812:4067
pyramid	2.0104	
Near—Best	1.6883	3 8 9 11 21 40 41
Best	1.6883	3 8 9 11 21 40 41

Sample 13	c27wvkn.asc	1396:1651
pyramid	2.0218	
Near—Best	1.5687	3 4 10 11
Best	1.5680	3 4 11 40 42 43 83 164 165

Sample 14	c27wvkn.asc	8920:9175
pyramid	1.9934	
Near—Best	1.6717	3 4 11 20 85 86 87 168 169
Best	1.6717	3 4 11 20 85 86 87 168 169

Sample 15	c27wvkn.asc	23317:23572
pyramid	1.7025	
Near—Best	1.4082	3 8 9 10 11
Best	1.4082	3 8 9 10 11

Sample 16	c27wvkn.asc	16393:16648
pyramid	1.6923	
Near—Best	1.2221	3 5 8 9
Best	1.2214	5 8 9 13 24 28 31
		51 61 100 101 117 118 119 120 232 233 242 243

Sample 17	c27wvpx.asc	11092:11347
pyramid	1.9173	
Near—Best	1.8725	3 9 11 17 20 21 32
		132 133 134 135
Best	1.8705	9 11 13 15 17 32 43 48
		51 80 82 85 100 101 114 117 119
		132 133 134 135 162 163 166 167 168 169
		196 197 198 199 224 225 226 227
		230 231 232 233 236 237

Sample 18	c27wvpx.asc	11342:11597
pyramid	1.9750	
Near—Best	1.9576	3 9 11 17 21 32 33 40 82 166 167
Best	1.9327	3 9 11 17 21 65 81 82
		128 129 132 133 134 135 160 161 166 167

Sample 19	c27wvpx.asc	4093:4348
pyramid	1.9015	
Near—Best	1.8283	3 8 9 11 20 21
Best	1.8263	8 9 11 12 13 15 21 28 80
		118 119 162 163 164 165 166 167
		232 233 234 235

Sample 20	c27wvpx.asc	29089:29344
pyramid	1.9411	
Near—Best	1.9153	3 9 11 17 21 33 40 64 82 83 130 131
Best	1.8973	3 9 11 17 21 40 64 82 83 130 131
		132 133 134 135

The Near—Best—Basis and the Best Basis for samples 1, 2, 17, 18 and 19, being the most interesting, are presented as Figures 4.9 to 4.18. Conclusions are presented and summarized in Chapter 5.

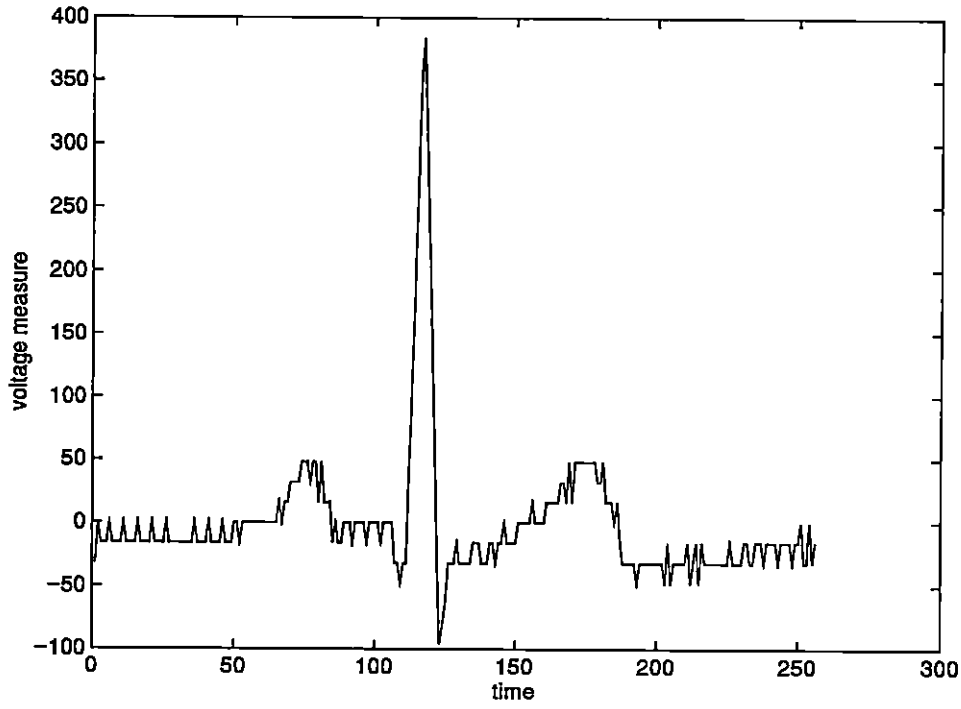


Figure 4.1: Sample 1

Time is in units of $\frac{1}{250}$ second. Voltage measure is particular to the ambulatory recording device being used and does not correspond exactly to a 12—lead ECG machine such as might be used in a hospital.

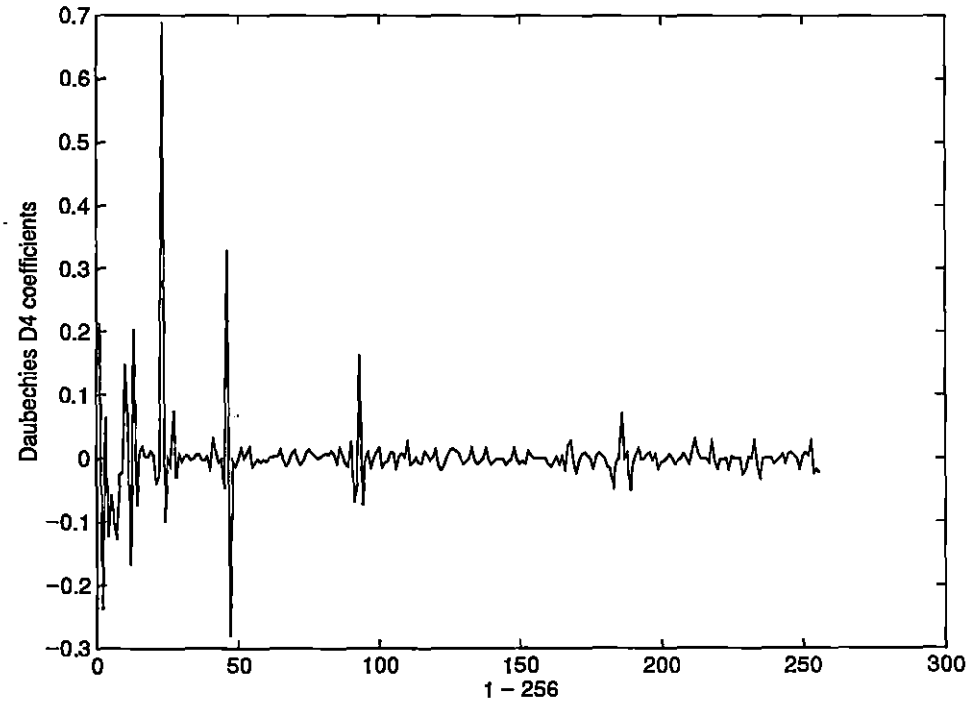


Figure 4.2: Daubechies D4 DWT of Sample 1

"wavel(ssc, daub4)" performs a discrete wavelet transform of the 256—vector ssc using the Daubechies D4 coefficients and the Pyramid algorithm.

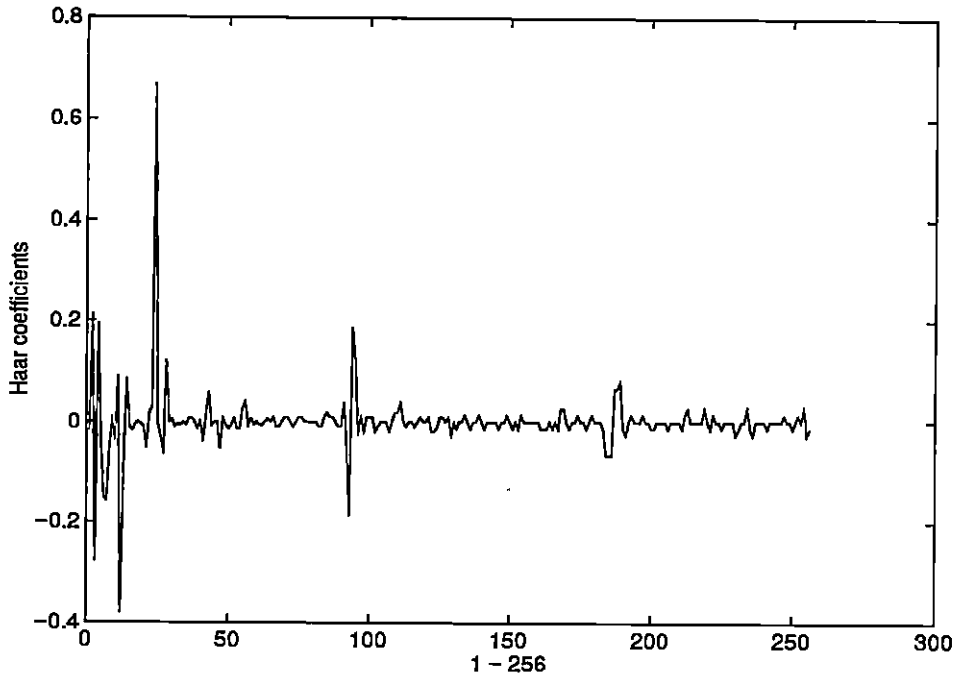


Figure 4.3: Haar DWT of Sample 1
"wavelhaar(ssc, haar2)" performs a discrete wavelet transform of the 256—vector ssc using the Haar coefficients and the Pyramid algorithm.

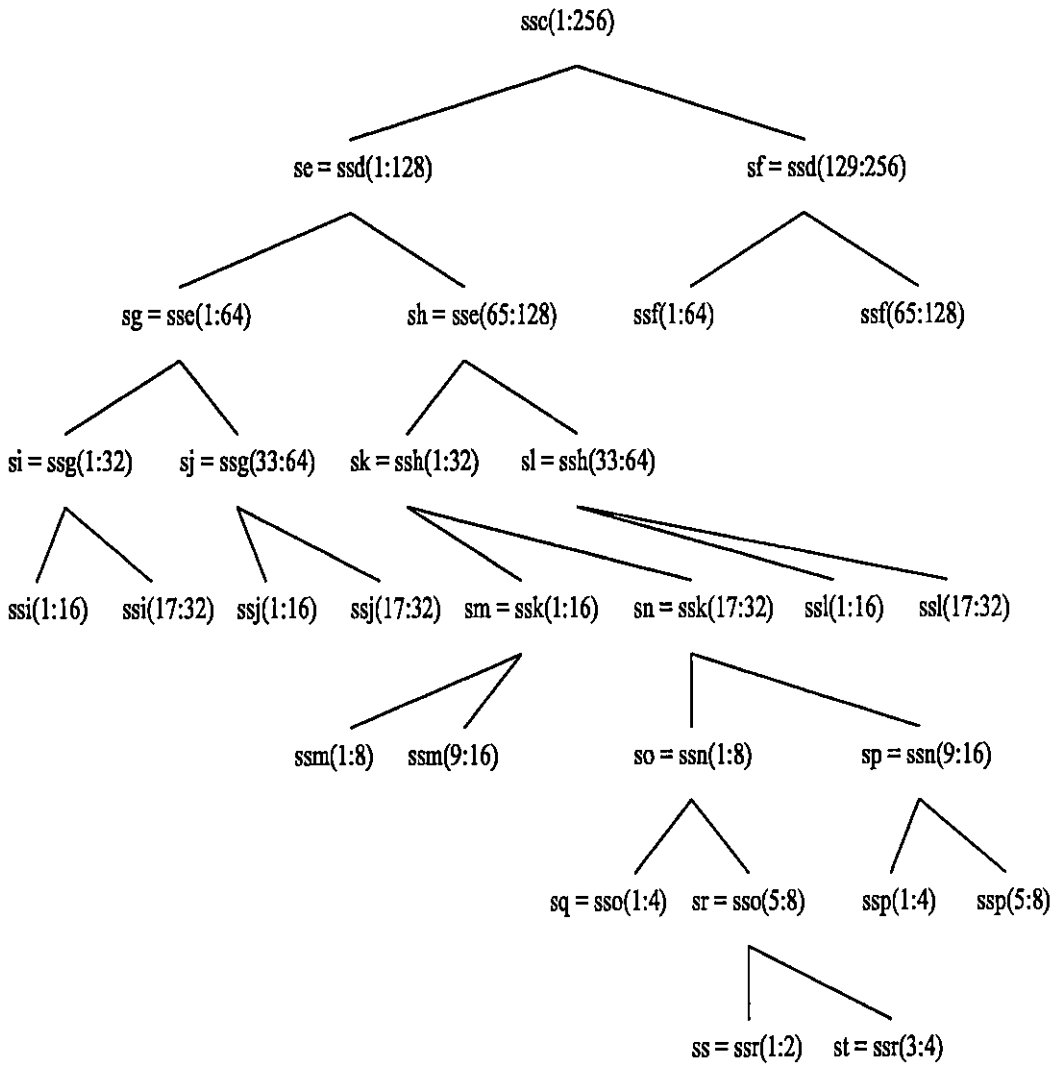


Figure 4.4: Near—Best—Basis Tree for Sample 1 (before pruning)
This tree is the result of all the operations carried out on Sample 1 in the first part of Chapter 4. The next illustration is the pruned version of this tree which constitutes the Near—Best—Basis for Sample 1.

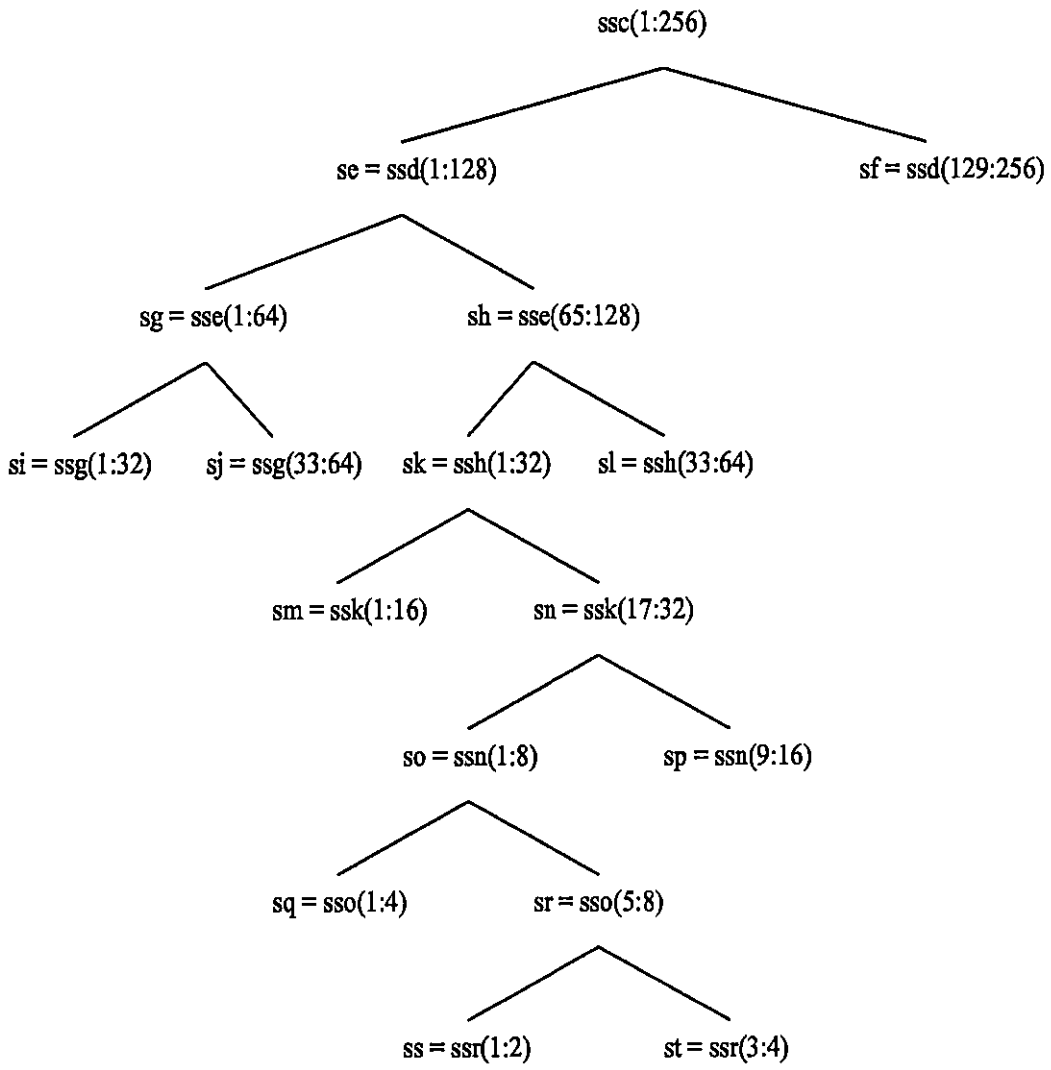


Figure 4.5: Near—Best—Basis Tree for Sample 1

This tree is the same as the previous tree, except it has been pruned to yield the Near—Best—Basis for Sample 1.

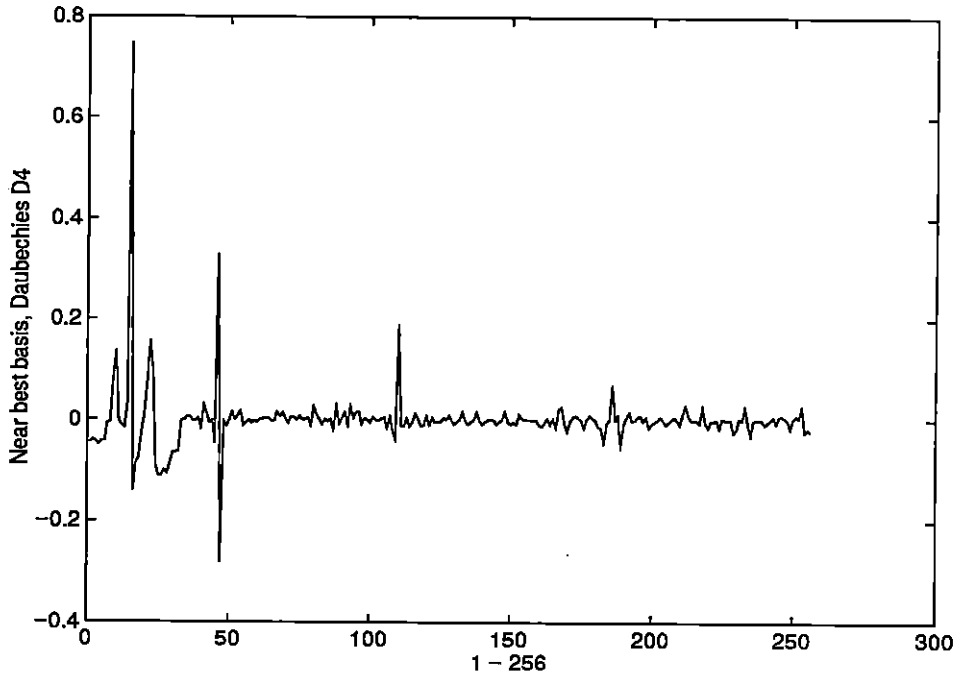


Figure 4.6: Near Best Basis Transform of Sample 1 using Daubechies D4 DWT

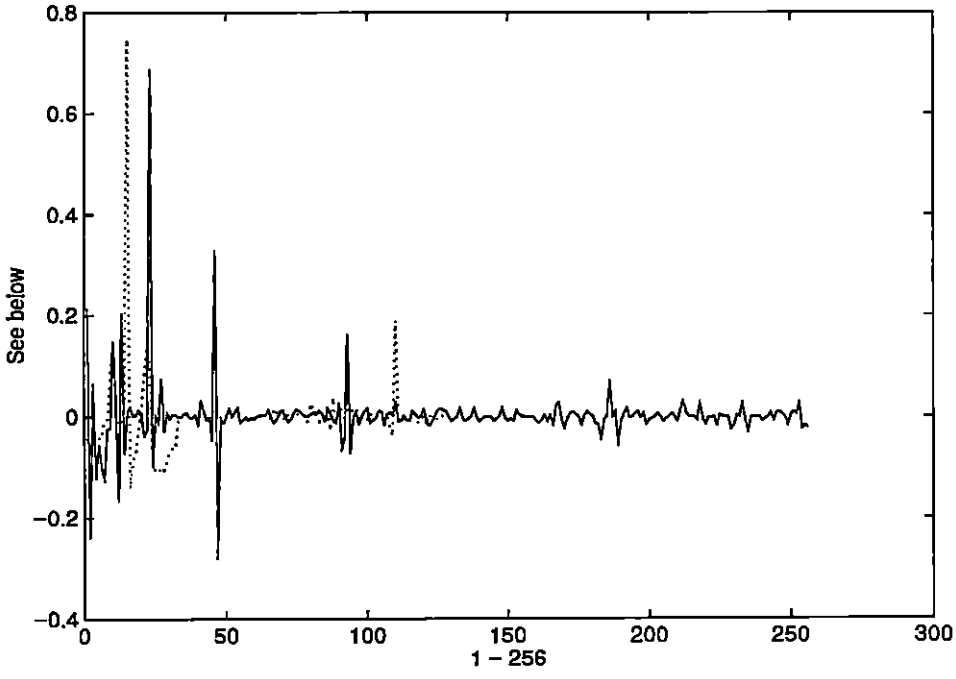


Figure 4.7: Comparison of Near Best Basis Transform of Sample 1 with Pyramid Algorithm, both using Daubechies D4 DWT
Pyramid algorithm is represented by a solid line while Near Best Basis is represented by dots

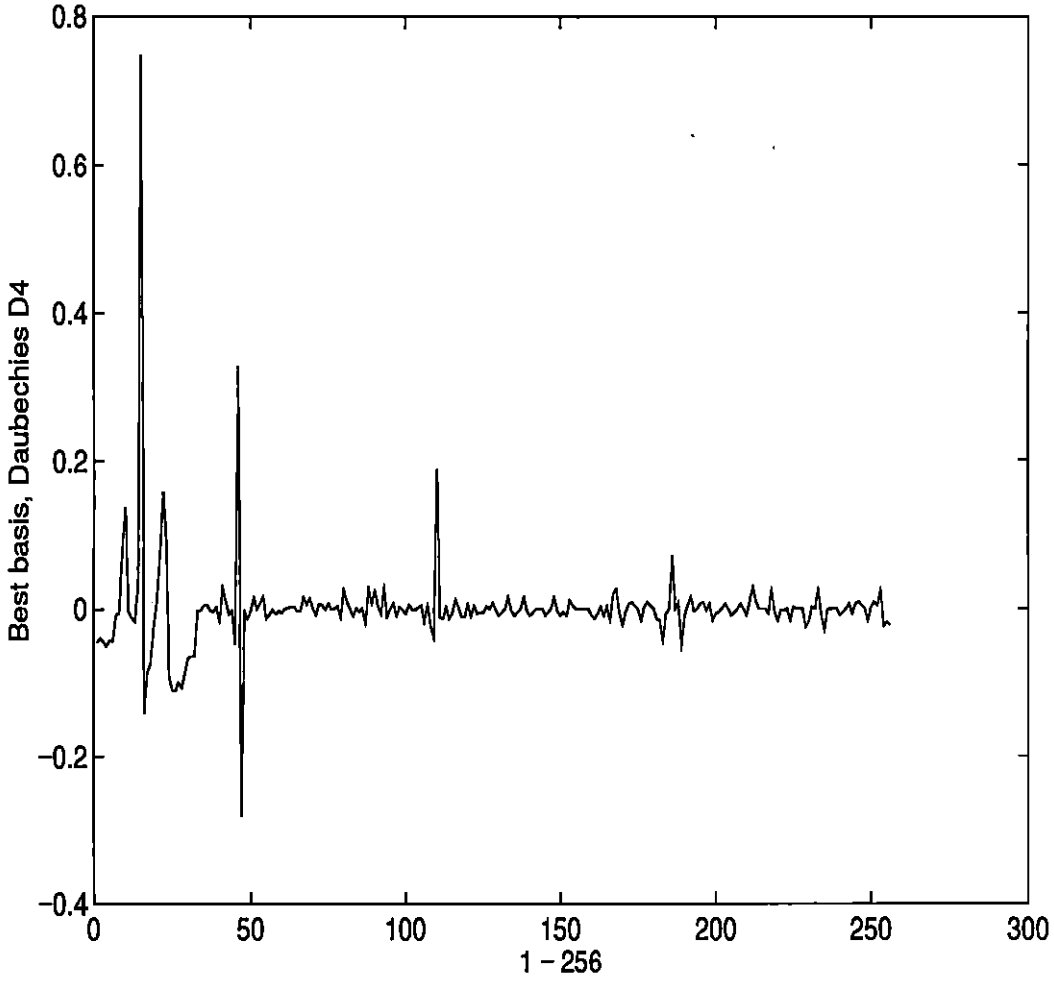


Figure 4.8: Best Basis Transform of Sample 1 using Daubechies D4 DWT

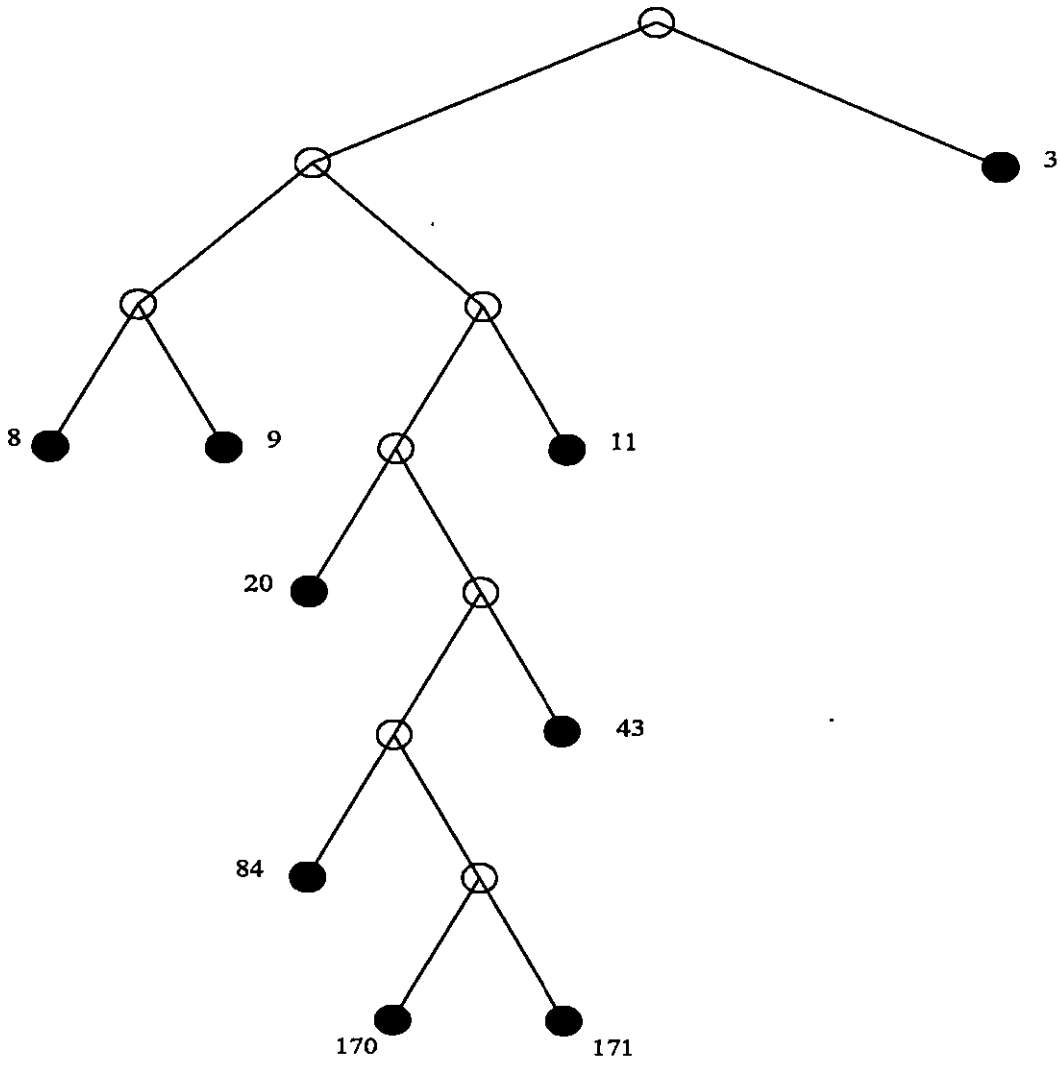


Figure 4.9: Sample 1: Near—Best—Basis

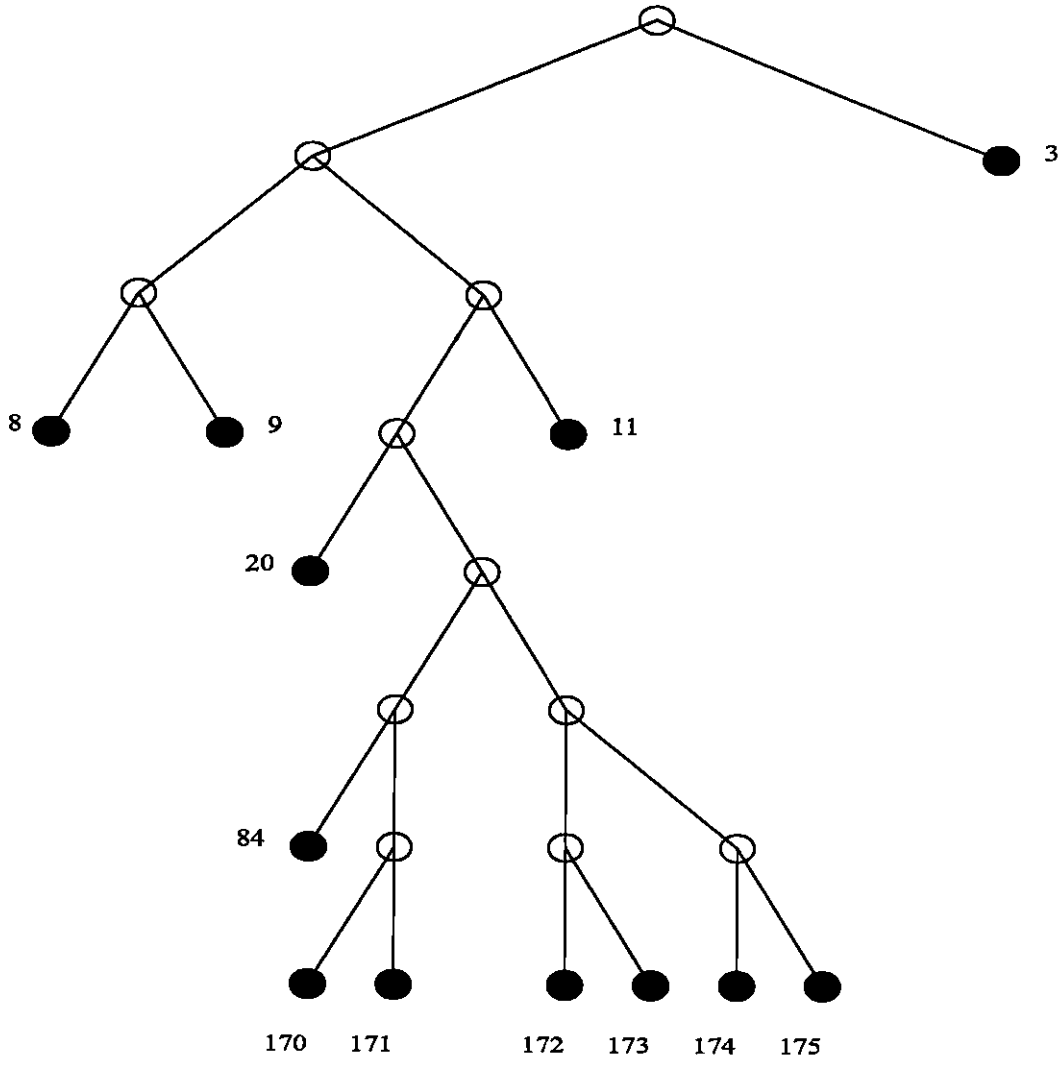


Figure 4.10: Sample 1: Best Basis

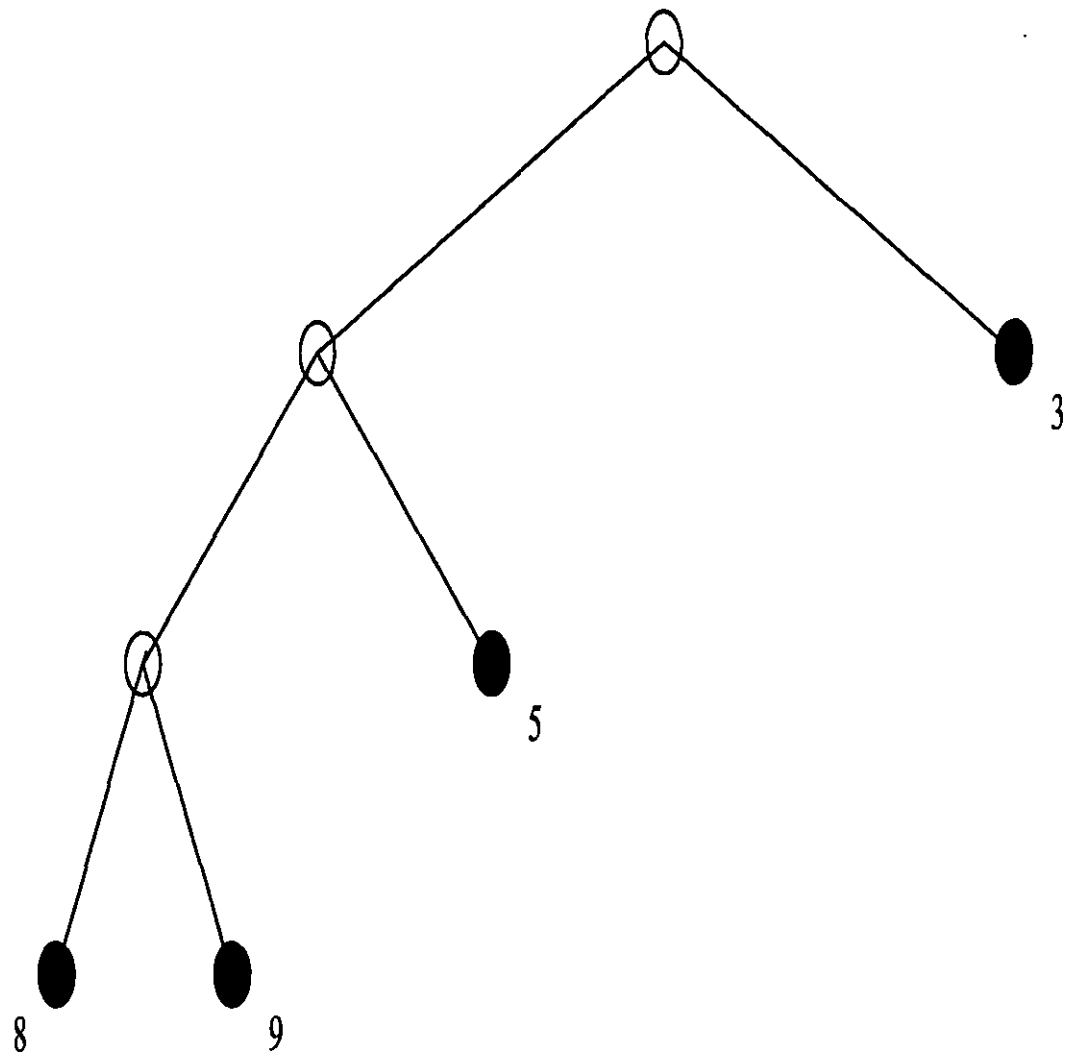


Figure 4.11: Sample 2: Near—Best—Basis

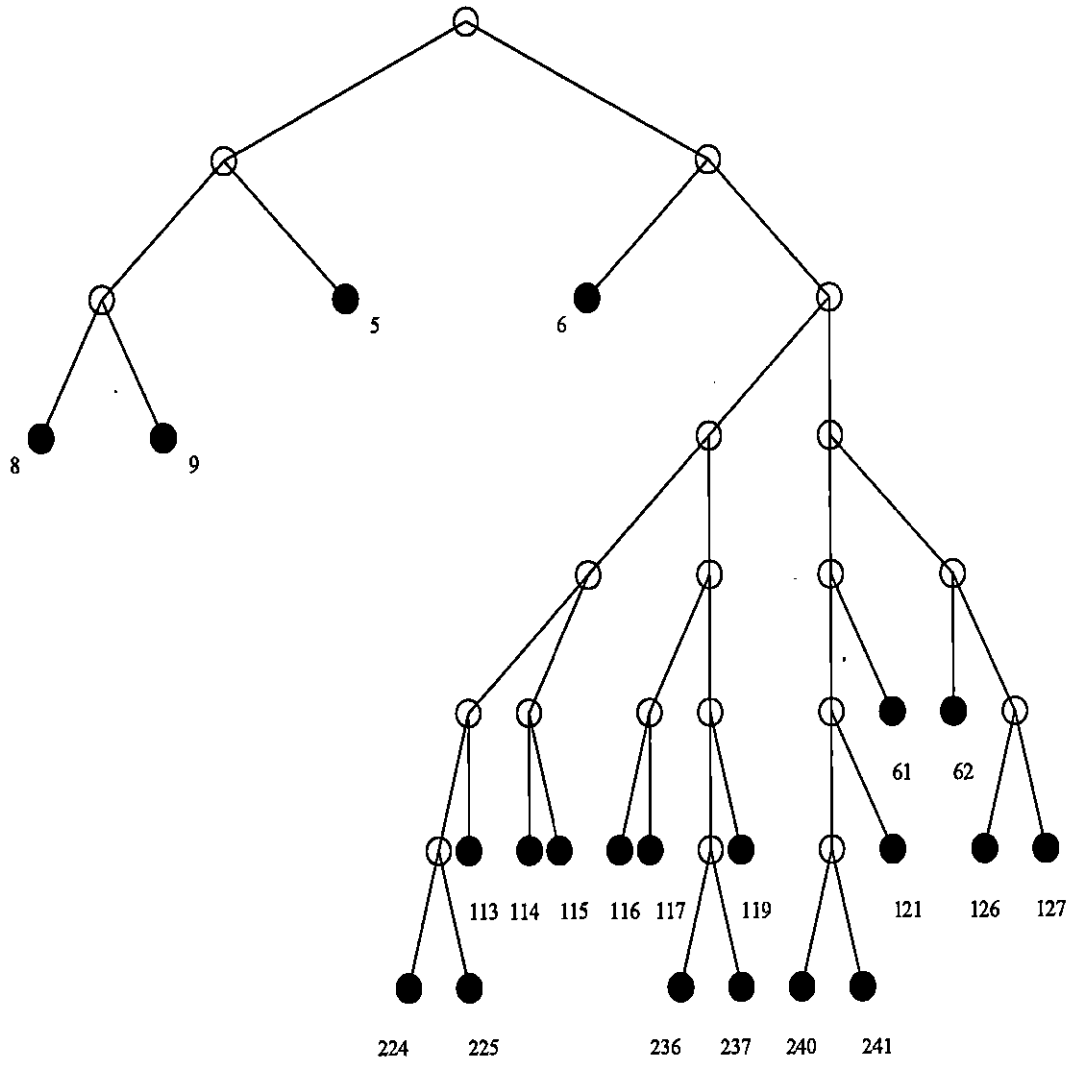


Figure 4.12: Sample 2: Best Basis

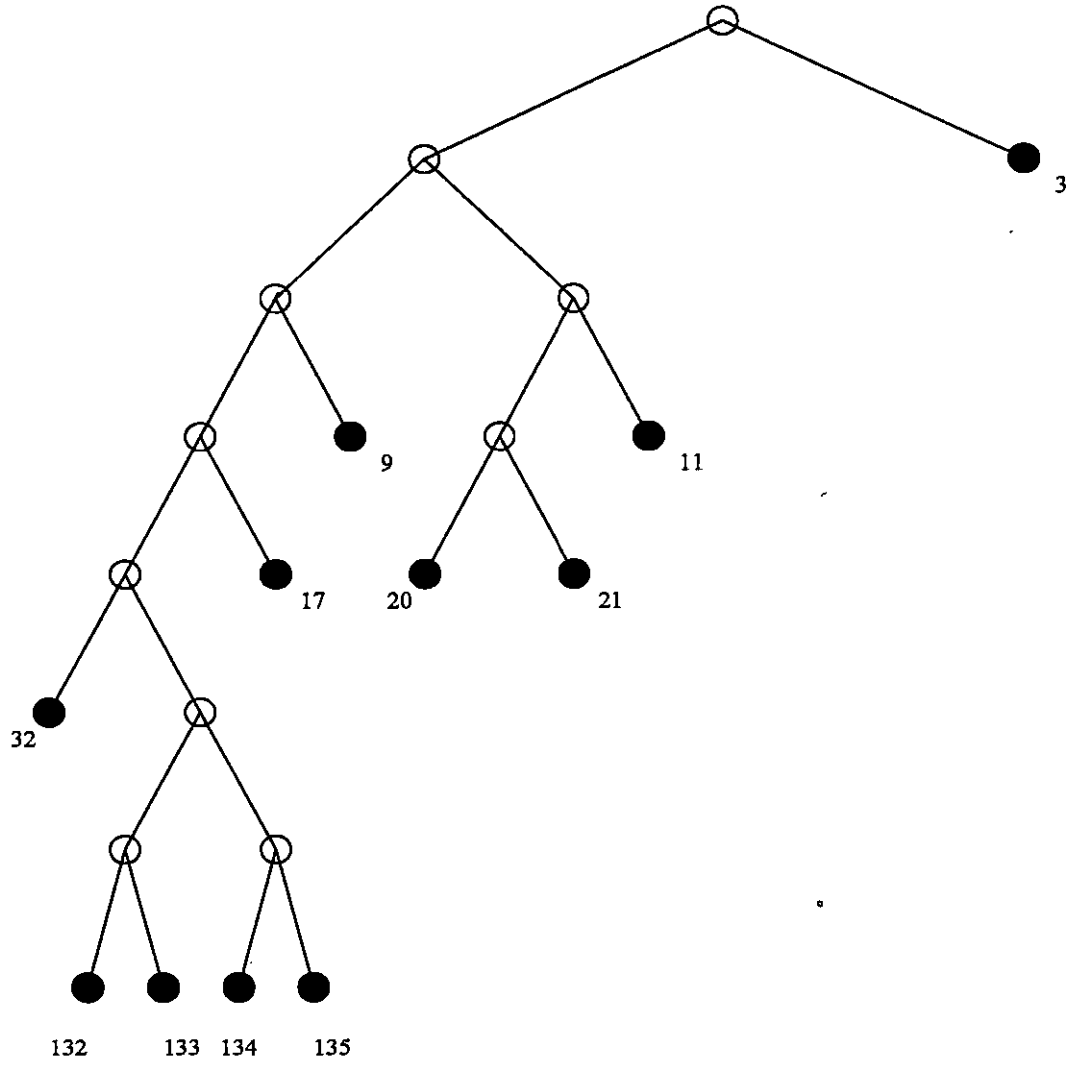


Figure 4.13: Sample 17: Near—Best—Basis

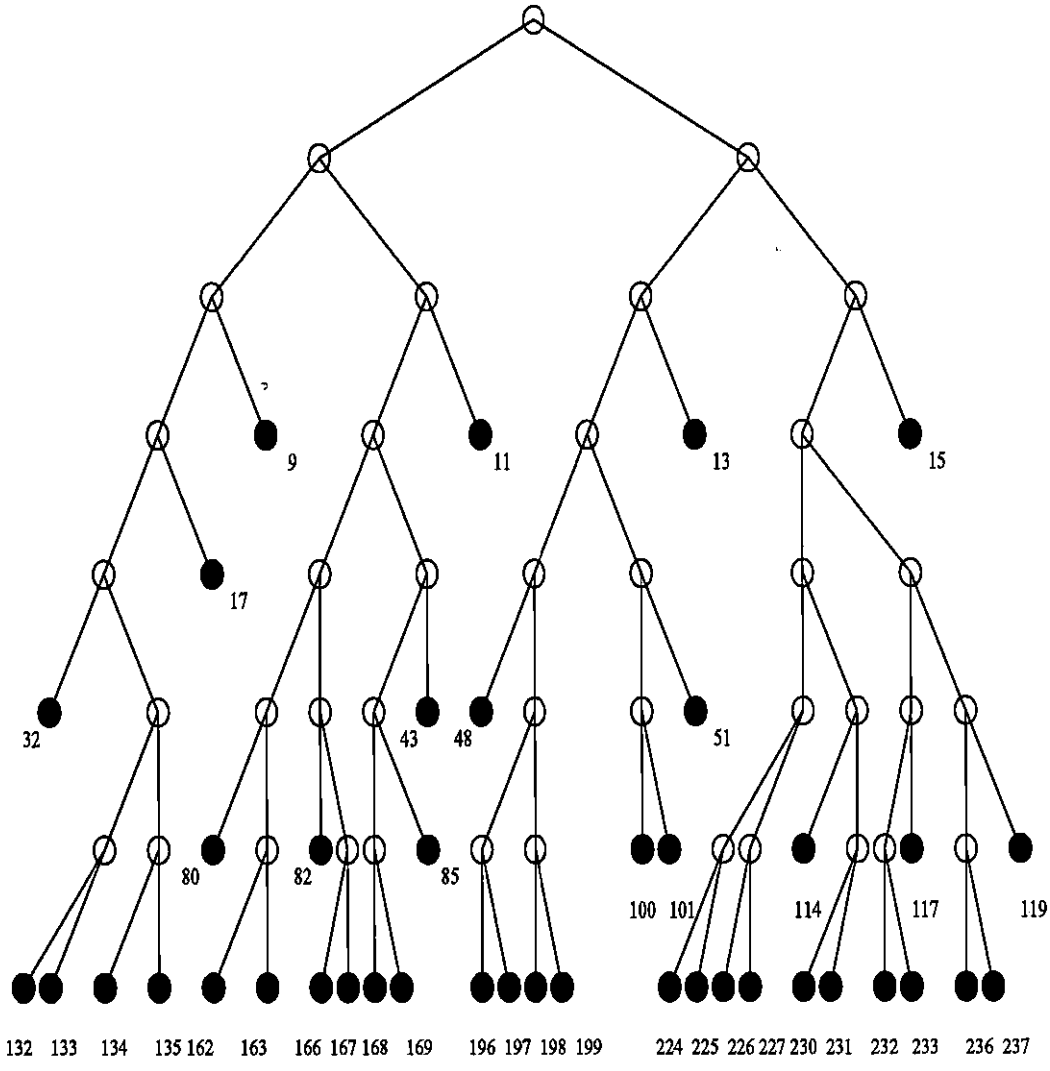


Figure 4.14: Sample 17: Best Basis

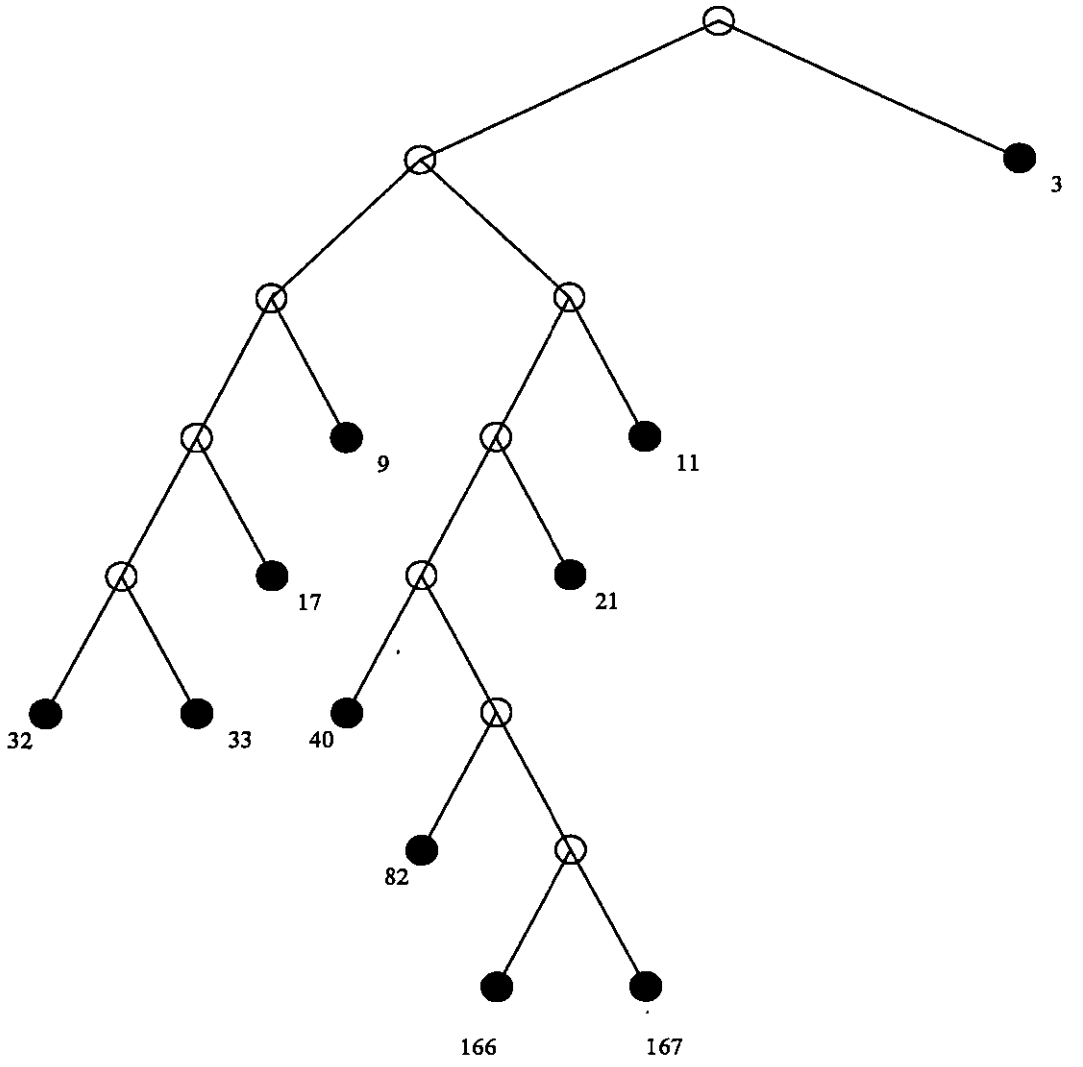


Figure 4.15: Sample 18: Near—Best—Basis

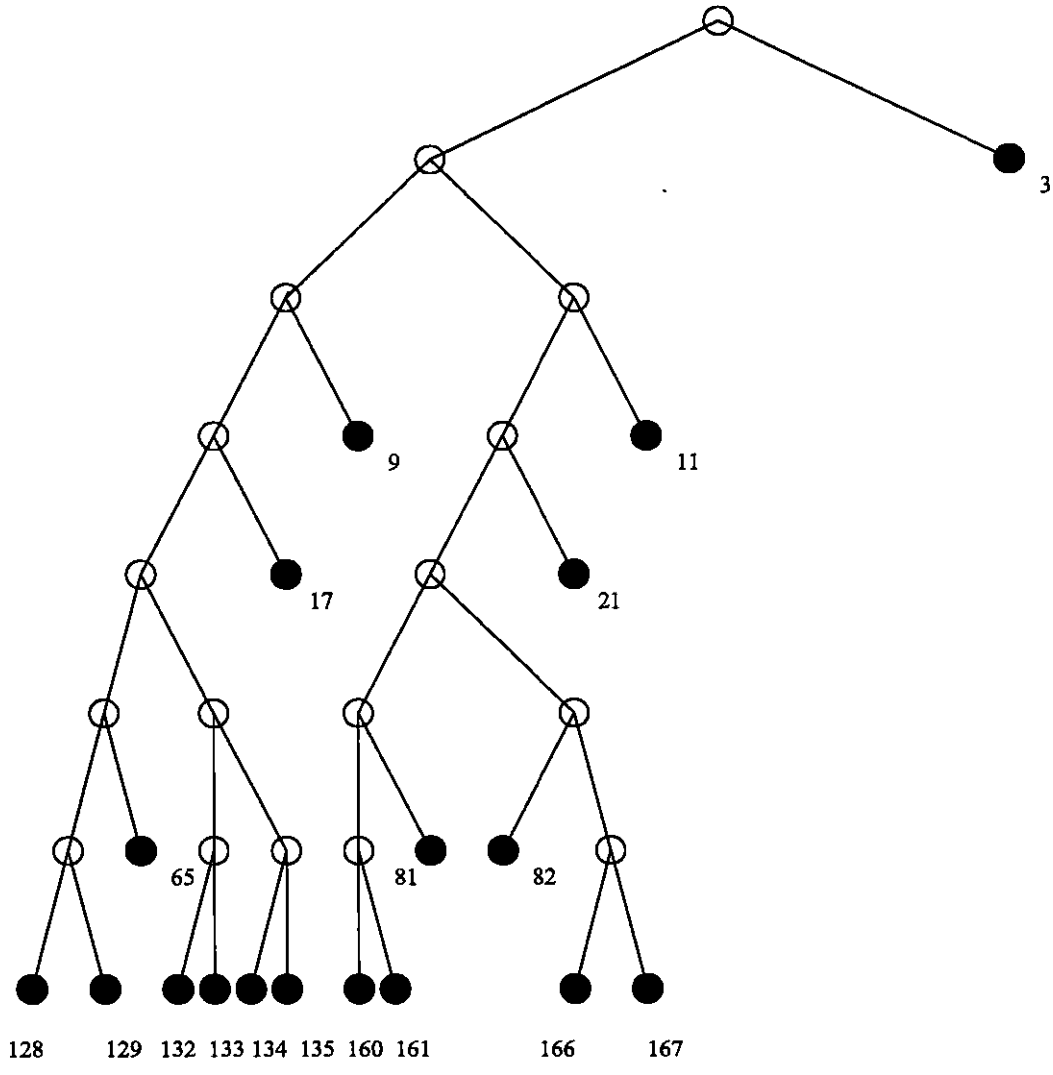


Figure 4.16: Sample 18: Best Basis

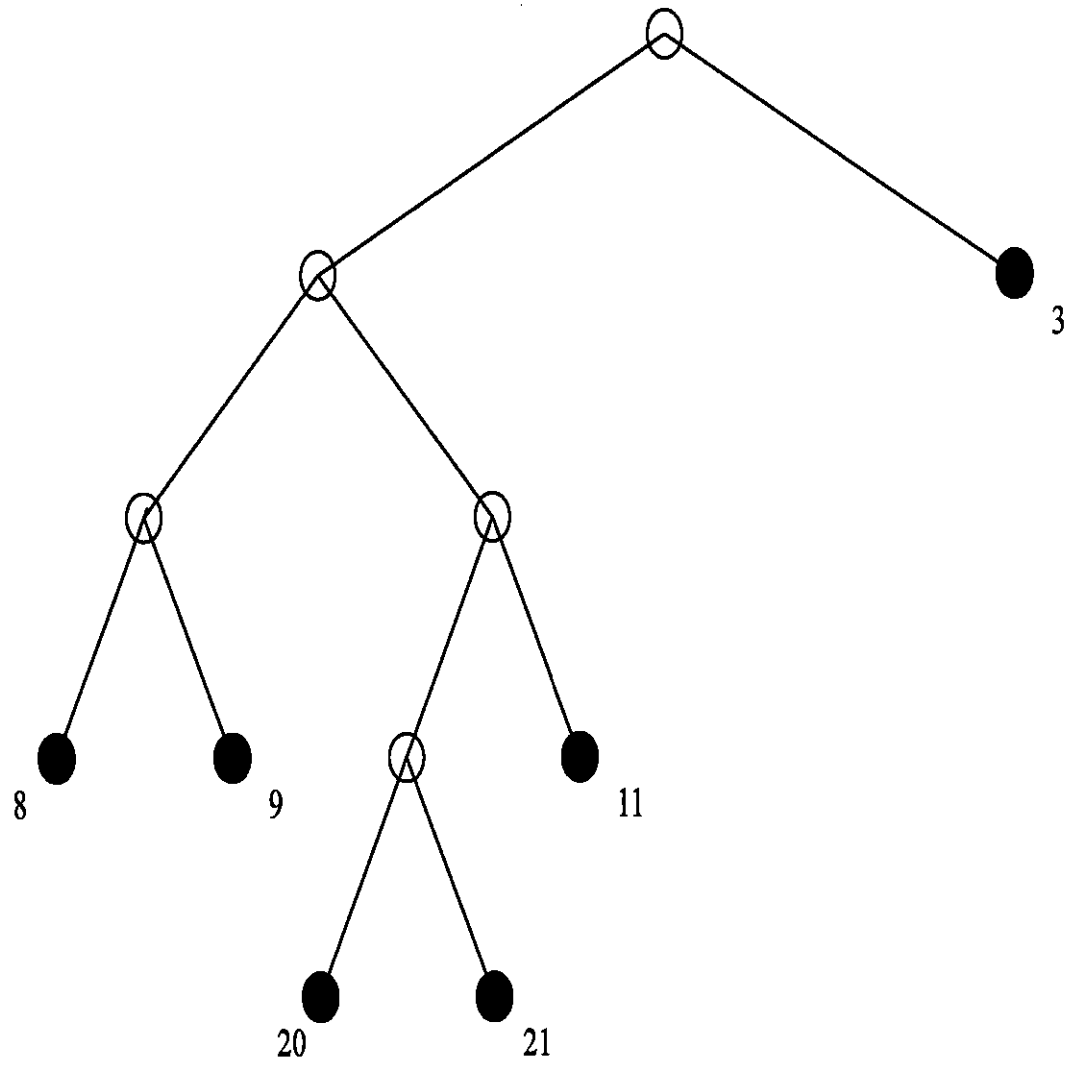


Figure 4.17: Sample 19: Near—Best—Basis

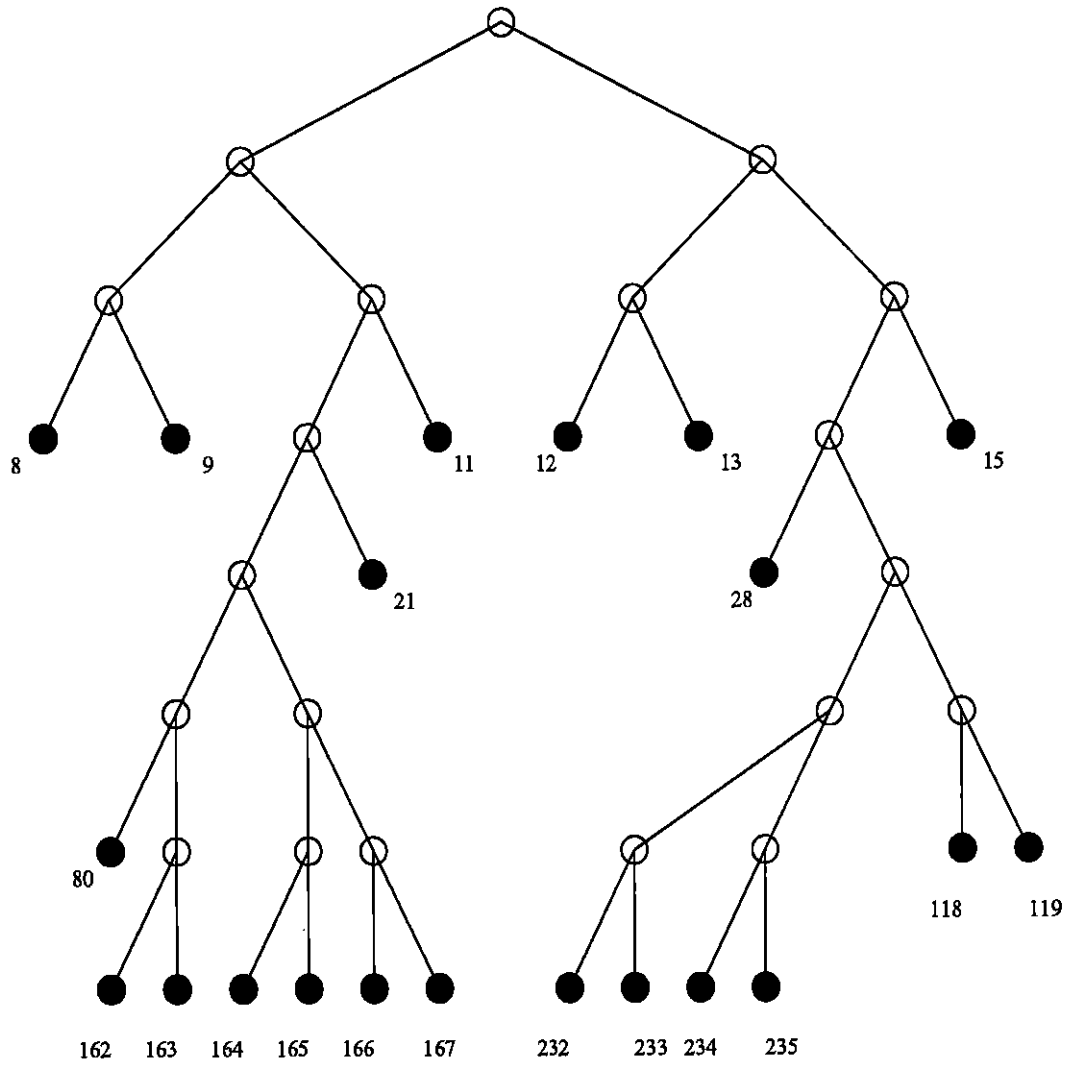


Figure 4.18: Sample 19: Best Basis

Chapter 5

Conclusions

5.1 Summary of Results

In general, the reduction in entropy which is achieved through the use of the Near—Best Basis is significant, and is larger than the further reduction which results from the use of the Best Basis.

The percentage reduction in entropy from Near—Best Basis compared to Pyramid Algorithm ranges from a low of 0.88% in Sample 18, to a high of 27.78% in Sample 16.

The percentage reduction in entropy from Best Basis compared to Near—Best Basis ranges from a low of 0% in 9 of the 20 samples, to a high of 1.27% in Sample 18.

Sample 18 bears closer examination, as it is unusual in two ways. It has the lowest reduction in entropy due to Near—Best Basis of the 20 samples, and the greatest reduction due to Best Basis compared to Near—Best Basis. It appears at the end of this Chapter, and appears to be a case of Atrial Flutter, a condition in which the upper chamber of the heart is out of step with the

lower chamber, in fact beating much more rapidly, resulting in numerous extra *p*—waves. There is no suggestion that a recording which produces numerical results similar to those of Sample 18 will be diagnosed similarly to Sample 18.

The Pyramid algorithm involves computing the contents of nodes 3, 5, 9, 17, 33, 65, 128 and 129, a total of 8 nodes. Of course, on the way to these nodes we also calculate the contents of nodes 2, 4, 8, 16, 32, and 64, another 6 nodes for a total of 14.

The Near—Best Basis involved computing anywhere from a low of 14 nodes, e.g. Sample 2 and Sample 4, to a high of 38 nodes, e.g. Sample 5. Consequently the conclusion is that the Near—Best Basis produces improved results over the Pyramid algorithm at a modest increase in computational cost.

The Best Basis always involves computing the contents of all 255 nodes prior to its determination. The conclusion is that the Best Basis produces marginal improvement over the Near—Best Basis, at a considerable increase in computational cost.

The final conclusion is that of the three methods, Pyramid algorithm, Near—Best Basis and Best Basis, the one to be preferred for this particular application is the Near—Best Basis of Taswell.

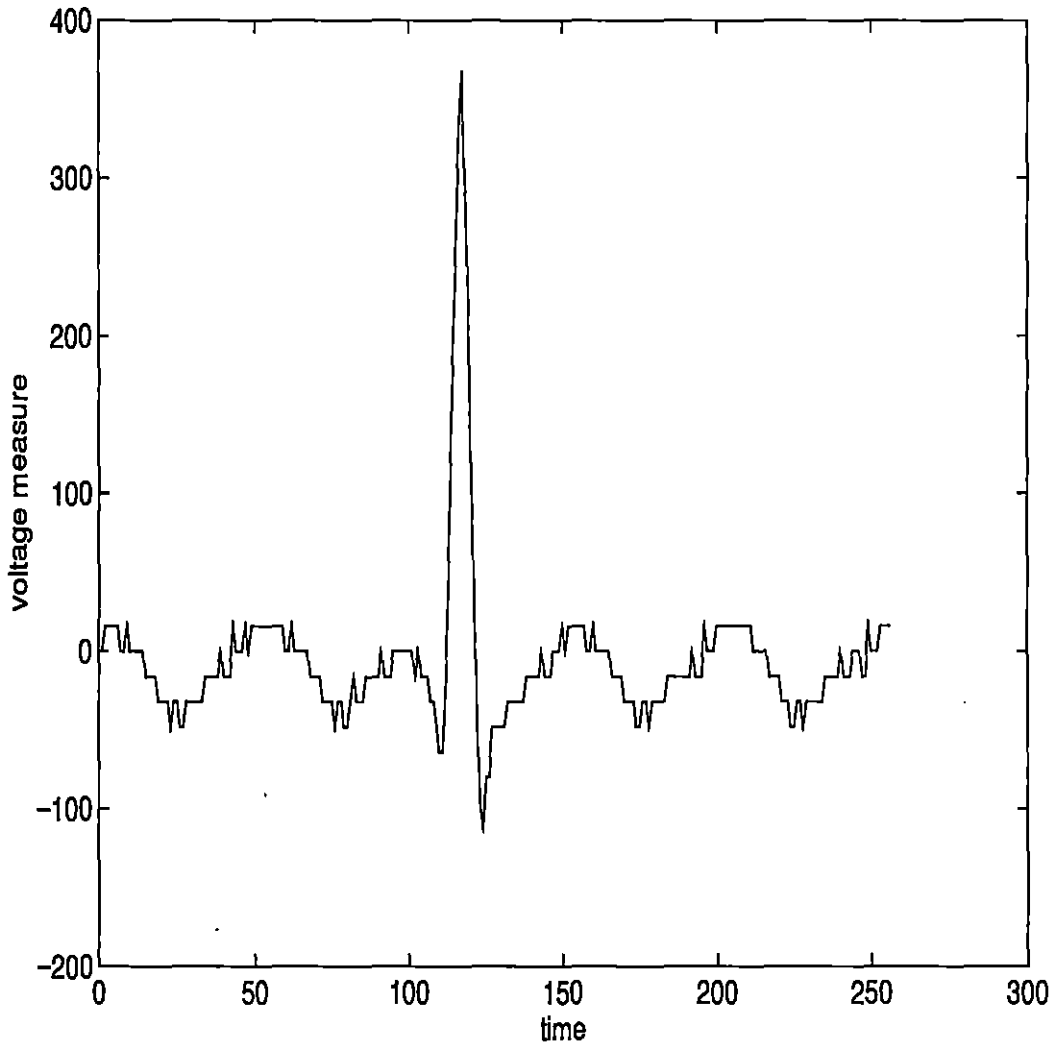


Figure 5.1: Sample 18

This sample has the lowest reduction in entropy due to Near—Best Basis of the 20 samples, and the greatest reduction due to Best Basis compared to Near—Best Basis.

Bibliography

[1] Benedetto, J.J. & Frazier, M.W. (1994). *Wavelets: Mathematics and Applications.* Boca Raton: CRC Press.

[2] Coifman, Ronald R. & Wickerhauser, Mladen V. (1990). *Best—adapted wave packet bases*, preprint, Yale University, New Haven.

[3] Coifman, Ronald R. & Wickerhauser, Mladen V. (1992). *Entropy—based Algorithms for Best Basis Selection.* IEEE Transactions on Information Theory, **38**, 2, 713-718.

[4] Cormen, Thomas H., Leiserson, Charles E. & Rivest, Ronald L. (1990). *Introduction to Algorithms.* Cambridge, Massachusetts: The MIT Press.

[5] Daubechies, Ingrid (1992). *Ten Lectures on Wavelets*, Philadelphia: Society for Industrial and Applied Mathematics.

[6] Daubechies, Ingrid (1988). *Orthonormal Bases of Compactly Supported Wavelets.* Communications on Pure and Applied Mathematics, **XLI**, 909-996.

[7] Dunford, N. & Schwartz, J. T. (1958). *Linear Operators, Part I: General Theory*, New York: Wiley–Interscience.

[8] Weidmann, J. (1980). *Graduate Texts in Mathematics. Vol. 68: Linear Operators in Hilbert Spaces.* New York: Springer–Verlag.

[9] Oppenheim, A. & Willsky, A. (1983). *Signals and Systems.* Englewood Cliffs, N.J.: Prentice—Hall, Inc..

[10] Press, W.H., Teukolsky, S.A., Vetterling, W.T. & Flannery, B.P. (1992). *Numerical Recipes in C.* Cambridge University Press.

[11] Ramirez, R.W. (1985). *The FFT, fundamentals and concepts.* Englewood Cliffs, N.J.: Prentice—Hall, Inc..

[12] Scheidt, Stephen, MD. *Basic Electrocardiography* Ciba—Geigy Pharmaceuticals internal publication.

[13] Strang, G. & Nguyen, T. (1996). *Wavelets and Filter Banks.* Wellesley: Wellesley—Cambridge Press.

[14] Strang, G. (1989). *Wavelets and Dilation Equations: a Brief Introduction.* SIAM Review, **31**, 4, 614-627.

[15] Taswell, Carl (1995). *Satisficing Search Algorithms for Selecting Near—Best Bases in Adaptive Tree—Structured Wavelet Transforms,* IEEE Transactions on Signal Processing preprint.

Appendix A

Application to Noise

We begin by considering a binary tree with only three nodes. Following usual convention, Node 1 is the parent node, and Nodes 2 and 3 are its left and right children, respectively. Now we consider a two—vector (x, y) such that x and y are independent identically distributed real variables. x and y are each normally distributed,with $\mu = 0$ and $\sigma = 1$. We may regard each such two—vector as being pure Gaussian noise.

Note that the Best Basis and the Near—Best—Basis representation of this vector using this tree will be identical. This is because a bottom—up and a top—down search of the tree will both consist of only one comparison, that of the parent with its two children. There are only two possibilities; either the parent node or the two children will be selected. Now the question is, using the Haar basis, what is the probability that the Best Basis representation will select the parent node?

Using the Haar basis, the parent vector is (x, y) and its two children are $\frac{x+y}{\sqrt{2}}$ and $\frac{x-y}{\sqrt{2}}$. The $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional of the parent is $-x^2 \log x^2 - y^2 \log y^2$. For the children, it is $-(\frac{x+y}{\sqrt{2}})^2 \log(\frac{x+y}{\sqrt{2}})^2 - (\frac{x-y}{\sqrt{2}})^2 \log(\frac{x-y}{\sqrt{2}})^2$. Note that replacing

x by $-x$ and/or y by $-y$ in either of these expressions does not alter their value. This means that the problem is symmetric about both the x-axis and the y-axis. Consequently we can restrict our attention to the first quadrant of the Cartesian plane and extend our result by this symmetry to the entire plane.

Since the x-axis and the y-axis are of measure zero, we can consider only points interior to the first quadrant. Now consider lines in the interior of the first quadrant which pass through the origin. Every point of each line will have the same Best Basis representation, since the first step is to normalize it. Consequently we need only consider points interior to the first quadrant and lying on the line $x = 1$. These will be of the form $(1, y)$ with $0 < y < \infty$. We now solve for those points on this line such that the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional of the parent and of its two children will be the same.

$$-y^2 \log y^2 = -\frac{(1+y)^2}{2} \log \frac{(1+y)^2}{2} - \frac{(1-y)^2}{2} \log \frac{(1-y)^2}{2} \quad \text{(A.1)}$$

$$= -\frac{(1+y)^2}{2} \log(1+y)^2 + \frac{(1+y)^2}{2} \log 2$$

$$-\frac{(1-y)^2}{2} \log(1-y)^2 + \frac{(1-y)^2}{2} \log 2 \quad \text{(A.2)}$$

$$= -(1+y)^2 \log(1+y) - (1-y)^2 \log|1-y| + \frac{(1+y)^2 + (1-y)^2}{2} \log 2 \quad \text{(A.3)}$$

$$= -(1+y)^2 \log(1+y) - (1-y)^2 \log|1-y| + (1+y^2) \log 2 \quad \text{(A.4)}$$

At this point a Hewlett—Packard HP48G was used to obtain numerical results, which led to the observation that the values $y = \sqrt{2}-1$ and $y = \sqrt{2}+1$ are both exact solutions.

The first y—value is $\tan \frac{\pi}{8}$ while the second is $\tan \frac{3\pi}{8}$. The final result is that if a point in the interior of the first quadrant, and by symmetry in any of

the four quadrants, lies on a ray which is within $\frac{\pi}{8}$ of either the x—axis or the y—axis, then either the Best or the Near—Best Basis will select the parent. If the ray is further away than that, the children will be selected.

Now by symmetry, one—quarter of the points will lie in the first quadrant. We are interested in the probability that a randomly selected point will have the two children selected. This equals

$$\frac{1}{\sqrt{2\pi}}\frac{1}{\sqrt{2\pi}}\int_0^{\infty}\int_{(\sqrt{2}-1)x}^{(\sqrt{2}+1)x} e^{-\frac{x^2}{2}}e^{-\frac{y^2}{2}}\,dy\,dx \tag{A.5}$$

now switching to Polar co—ordinates

$$= \frac{1}{2\pi}\int_{\frac{\pi}{8}}^{\frac{3\pi}{8}}\int_0^{\infty} e^{-\frac{r^2}{2}}\,r\,dr\,d\Theta \tag{A.6}$$

$$\tag{A.7}$$

now substituting $u = -\frac{r^2}{2}$

$$= \frac{1}{2\pi}\int_{\frac{\pi}{8}}^{\frac{3\pi}{8}}\int -e^{-u}\,du\,d\Theta \tag{A.8}$$

$$= \frac{1}{2\pi}\int_{\frac{\pi}{8}}^{\frac{3\pi}{8}} -e^{-\frac{r^2}{2}}\,|_0^{\infty} d\Theta \tag{A.9}$$

$$= \frac{1}{2\pi}\int_{\frac{\pi}{8}}^{\frac{3\pi}{8}} d\Theta \tag{A.10}$$

$$= \frac{1}{2\pi}\Theta|_{\frac{\pi}{8}}^{\frac{3\pi}{8}} \tag{A.11}$$

$$= \frac{1}{8} \tag{A.12}$$

Consequently, for exactly half the points in the first quadrant, the algorithm will select the children over the parent. By symmetry, this result extends to the entire plane. We conclude that if data consists of such pure

noise two-vectors, then this implies that the algorithm will select the parent with probability 0.5, and the children with the same probability. Of course, we have not proved the converse; that if the selection is equiprobable for parent and children, then the signal is pure noise.

We now present some empirical evidence that this result may be regarded as holding for 4—vectors, using both the Haar and the DAUB4 wavelets.

First, using MATLAB, we generate 10,000 4—vectors, each entry a normally distributed independent random variable with $\mu = 0$ and $\sigma = 1$. Using a three—node tree as above, the Haar wavelet and the Best Basis, (although Near—Best—Basis would have given exactly the same result), we present results for five such samples. This is repeated using the Daubechies D4 wavelet. The results are presented on page 86.

It is readily apparent that for practical applications we may consider that pure Gaussian noise will have a Best Basis representation as the parent, or as the two children, with equal probability.

One consequence of this is that if we were to consider a 256—vector of pure noise, with each entry a normally distributed independent random variable with $\mu = 0$ and $\sigma = 1$, then the Near—Best—Basis using the full algorithm "treat" would be Node 1 with a probability of 0.5, since the search is from the top down. The Best Basis of the signal would contain numerous nodes, since the search is bottom up, and at each step there is an equal probability of accepting the children or the parent.

As was mentioned in Chapter 4, the further treatment of the heartbeats, the deletion of small coefficients to achieve de—noising and data compression, remains beyond the scope of this thesis. However, here we will permit the

comment that the above observations on noise will be useful in refining the de—noising process. We can always subtract the de—noised signal from the original signal to examine the supposed "noise" which has been removed. Now using a three—node tree, we can perform statistical checks on the "noise". So long as the number of times the "noise" is associated with the parent as opposed to the children is consistent with a coin tossing experiment, then an experimenter could be reasonably sure that what was removed is, indeed, primarily noise. If the number is inconsistent, say if it is more than three standard deviations above or below the mean, then the experimenter may well suspect that the amount of the "energy" of the signal which is being removed is too high. That is, we are losing valid signal content along with the noise. This would mean reducing the level of "energy" being removed.

Haar wavelet

Trial number	Parent selected	Children selected
1	4982	5018
2	5042	4958
3	4963	5037
4	4987	5013
5	5080	4920

Table A.1: Experiment using Haar wavelet

DAUB4 wavelet

Trial number	Parent selected	Children selected
1	5011	4989
2	4960	5040
3	4952	5037
4	4987	5048
5	5027	4973

Table A.2: Experiment using Daub4 wavelet

Appendix B

Computer Programs

What follows is a listing of the computer programs used in this thesis. With the exception of "wave" and "invwavel", all were written by the author.

Best

"Best" accepts as input the 256—vector which is the normalized version of the heartbeat, and outputs a 2303—vector. The first 2048 positions in the output are the eight levels of the complete decomposition of the signal into the binary tree $\mathbb{W}$ referred to in Section 3.3, using the Daubechies D4 wavelet. The last 255 positions are the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals calculated at each node. There is one exception to this, and that is the values associated with the lowest level of the tree are not calculated separately. The reason for this is that the first step to comparing the children with the parent is to add the entropy functionals of the two children together, so redundancy is avoided on the lowest level of the tree by not separating the values for the children.

Indic

"Indic" accepts as input the last 255 positions of the output of "Best" referred to immediately above. It performs the Best Basis selection of Coifman and

Wickerhauser and outputs a 510—vector. The first 255 positions of output are the modified values of entropy functional referred to in Section 3.3. The last 255 positions are the indicators referred to in the same section, "0" for a node which is "not kept" and "1" for a node which is "kept".

Invert

"Invert" is used in the reconstruction algorithm "recon". It accepts as input the data contained at two children of the same parent node. What is significant is that the data at the two nodes may already have been modified by some de—noising or data compression algorithm. The output is the reconstructed parent node, including the effect of the de—noising or data compression algorithm, using the Daubechies D4 wavelet.

Invwavel

This algorithm, which is essentially to be found in [10], takes as input the wavelet decomposition of a signal using the Pyramid algorithm, and the four coefficients of the Daubechies D4 wavelet. Its output is the inverse transform.

Invwavelent

"Invwavelent" is a modified version of "Invwavel". It performs the inverse wavelet transform on the output of "Wavelent", which decomposes a parent node into its two children and stores the values of the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals for the parent and the two children.

Invwavelenttrunc

"Invwavelenttrunc" is similar to "Invwavelent", except that it performs an inverse wavelet transform on two children to produce their parent without storage of the associated entropy functionals.

Invwavelhaar

"Invwavelhaar" is a modified version of "Invwavel" which uses the Haar basis. The functions "Wavel" and "Invwavel" as received could only be used with the Daubechies D4 to D20 coefficients, but not the Haar coefficients.

Loc

"Loc" is called by the function "Recon". It accepts as input the 256—vector resulting from any wavelet representation together with the results of any de—noising and data compression scheme, and a single integer from 1 to 255. It outputs those data values necessary for "recon".

Nb

"Nb" is called by "treat". It accepts as input the 255 values of the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals associated with each node of the binary tree $\mathbb{W}$, and performs the top—down search referred to in Section 3.4. It outputs the indicators for the Near—Best—Basis, "0" for "not kept" nodes, "1" for "kept" nodes and "-1" for descendants of "not kept" nodes. This last code ensures that there are no unnecessary repetitive steps.

Nbent

"Nbent" is called by "treat". It accepts as input the 255 values of the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals associated with each node of the binary tree $\mathbb{W}$, as well as the output of "Nb" above. From this, it calculates and outputs the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional associated with the Near—Best—Basis representation.

Pyr

"Pyr" is called by "treat". It accepts as input the 255 values of the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals associated with each node of the binary tree $\mathbb{W}$, and calculates and outputs the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional associated with the Pyramid algorithm representation.

Recon

"Recon" accepts as input any binary tree of "kept" and "not kept" nodes, together with any wavelet transform of the signal, and produces a reconstructed version of the original signal. As pointed out in Chapter 4, what is remarkable about "recon" is that it can reconstruct a signal which has been modified from its original wavelet representation by *any* denoising technique based on *any* of the possible subtrees, of which there are over 1.8×10^{19}.

Srt

"Srt" is called by "treat" twice, once to produce a list of the "kept" nodes in the Near—Best—Basis, and again for the Best Basis. It accepts as input the 255 values of the "kept" and "not kept" indicators and outputs a list of the positions of the "kept" nodes.

Stack

"Stack" is called by "wave". It is similar to "Loc" except that instead of accepting a de—noised and data compressed version of a wavelet transform, it accepts the 2048 positions which are the eight levels of the complete decomposition of the signal into the binary tree $\mathbb{W}$ referred to in Section 3.3, using the Daubechies D4 wavelet. This is then combined with a single integer from 1 to 255, and the output is the data at that node.

Treat

"Treat" accepts as input a 256—vector of raw data. It calls on subroutines "best", "pyr", "nb", "nbent", "srt", and "indic". It outputs the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functional associated with the Pyramid algorithm wavelet representation, followed by that associated with the Near—Best—Basis, followed by a listing of the "kept" nodes of the Near—Best—Basis, followed by the entropy

functional for the Best Basis, and lastly the "kept" nodes of the Best Basis.

Wave

"Wave" accepts as input the 2048 positions which are the eight levels of the complete decomposition of the signal into the binary tree $\mathbb{W}$ referred to in Section 3.3, using the Daubechies D4 wavelet. This is combined with the "kept" nodes of some wavelet represenation, whether Pyramid, Near—Best—Basis, Best Basis, or some scheme to be implemented in the future. It outputs a 256—vector which consists of the wavelet transform under the scheme being used.

Wavel

This algorithm, which is essentially to be found in [10], takes as input the 256—vector of raw data, and uses the Pyramid algorithm and the four coefficients of the Daubechies D4 wavelet. Its output is the resulting wavelet transform.

Wavelent

"Wavelent" is a modified version of "Wavel". Unlike the Pyramid algorithm, which goes all the way to the bottom level, it takes a parent and produces its two children, together with the stored values of the $\mathbb{L}^2 \log \mathbb{L}^2$ entropy functionals for the parent and for the two children.

Wavelenthaar

"Wavelenthaar" is a modified version of "Wavelent" which uses the Haar wavelet instead of the Daubechies D4 wavelet.

Wavelentnorm

"Wavelentnorm" is a modified version of "Wavelent" which accepts as input only raw data which has already been normalized. This permits faster calculation.

Wavelhaar

"Wavelhaar" is a modified version of "Wavel" which uses the Haar wavelet instead of the Daubechies D4 wavelet.

Wavelmod

"Wavelmod" is a modified version of "Wavel" which, instead of performing the Pyramid algorithm, only goes down one level. That is, it accepts as input a parent node and outputs its two children.

Vita

Surname: STEACY Given Names: ROBERT CLIFFORD BRUCE

Place of Birth: Ottawa, Ontario, Canada Date of Birth: June 1, 1952

Educational Institutions Attended:

University of Victoria	1993 to 1996
University of British Columbia	1968 to 1972

Degrees Awarded:

B.Sc. (Honours)	University of British Columbia	1972

Honours and Awards:

University of Victoria Graduate Fellowship	1994-1996
University of B.C.'s best paper, W.L. Putnam Mathematics Competition	1971
University of B.C.'s best paper, Canadian Association of Physicists Prize Examination	1971
General Motors of Canada Scholarship	1968-1972
Governor General's Bronze Medal, Province of B.C.	1968

Publications:

R. Steacy, *Structure in the Kolakoski Sequence.* Bulletin of the European Association for Theoretical Computer Science, **59**, 173—182. As DMS - 725 - IR January 1996 this paper was cited by Michel Dekking in The Mathematics of Aperiodic Order, Proc. NATO Adv. Studies Inst., Aug. 21 - Sept. 1, 1995 (Eds. R.V. Moody and J. Patera), Kluwer Academic Publishers,Dordrecht/Boston/London, 1996.

Partial Copyright License

I hereby grant the right to lend my thesis to users of the University of Victoria Library, and to make single copies only for such users or in response to a request from the Library of any other university, or similar institution, on its behalf or for one of its users. I further agree that permission for extensive copying of this dissertation for scholarly purposes may be granted by me or a member of the University designated by me. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Title of Dissertation:

Comparison of a Best and a Near—Best Basis Approach to Wavelet Analysis of Electrocardiograms

Author [redacted]

Robert Clifford Bruce Steacy
(Name)

September 13, 1996
(Date)